
AWS Security Hub

User Guide



AWS Security Hub: User Guide

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS Security Hub?	1
Benefits of Security Hub	1
Reduced effort to collect and prioritize findings	1
Automatic security checks against best practices and standards	1
Consolidated view of findings across accounts and providers	1
Ability to automate remediation of findings	1
How Security Hub works	1
Security Hub free trial, usage, and pricing	2
Viewing usage details and estimated cost	2
Viewing pricing details	3
Terminology and concepts	4
Prerequisites and recommendations	8
Using Organizations	8
Enabling AWS Config	8
How to enable AWS Config	9
Configuring resource recording in AWS Config	9
Quotas	11
Maximum quotas	11
Rate quotas	12
Regional limits	13
Cross-Region aggregation restrictions	13
Integrations not supported in all Regions	14
Integrations that are supported in China (Beijing) and China (Ningxia)	14
Integrations that are supported in AWS GovCloud (US-East) and AWS GovCloud (US-West)	14
Controls not supported in all Regions	15
US East (Ohio)	15
US West (N. California)	16
US West (Oregon)	17
Africa (Cape Town)	18
Asia Pacific (Hong Kong)	21
Asia Pacific (Jakarta)	22
Asia Pacific (Mumbai)	29
Asia Pacific (Osaka)	30
Asia Pacific (Seoul)	34
Asia Pacific (Singapore)	34
Asia Pacific (Sydney)	35
Asia Pacific (Tokyo)	36
Canada (Central)	37
China (Beijing)	37
China (Ningxia)	42
Europe (Frankfurt)	47
Europe (Ireland)	47
Europe (London)	48
Europe (Milan)	49
Europe (Paris)	52
Europe (Stockholm)	53
Middle East (Bahrain)	54
South America (São Paulo)	55
AWS GovCloud (US-East)	56
AWS GovCloud (US-West)	61
Setting up Security Hub	66
Enabling Security Hub manually	66
Attaching the required IAM policy to the IAM identity	67
Enabling Security Hub (console)	67

Enabling Security Hub (Security Hub API, AWS CLI)	67
Enabling Security Hub (Multi-account script)	68
Service-linked role assigned to Security Hub	68
Security	69
Data protection	69
AWS Identity and Access Management	70
Audience	70
Authenticating with identities	71
AWS account root user	71
IAM users and groups	71
IAM roles	72
Managing access using policies	72
How AWS Security Hub works with IAM	74
Using service-linked roles	80
Service-linked role permissions for Security Hub	80
Creating a service-linked role for Security Hub	81
Editing a service-linked role for Security Hub	81
Deleting a service-linked role for Security Hub	81
AWS managed policies	81
AWSSecurityHubFullAccess	82
AWSSecurityHubReadOnlyAccess	82
AWSSecurityHubOrganizationsAccess	83
AWSSecurityHubServiceRolePolicy	84
Policy updates	85
Compliance validation	86
Infrastructure security	86
VPC endpoints (AWS PrivateLink)	87
Considerations for Security Hub VPC endpoints	87
Creating an interface VPC endpoint for Security Hub	87
Creating a VPC endpoint policy for Security Hub	87
Managing accounts	89
Effects of an administrator-member relationship	89
Restrictions and recommendations	90
Maximum number of member accounts	90
Accounts and Regions	90
Restrictions on administrator-member relationships	90
Coordinating administrator accounts across services	90
Making the transition to Organizations	91
Designate a Security Hub administrator account for your organization	91
Enable organization accounts as member accounts	92
Allowed actions for accounts	92
Designating a Security Hub administrator account	93
How the Security Hub administrator account is managed	93
Required permissions to configure the Security Hub administrator account	95
Designating a Security Hub administrator account (console)	95
Designating a Security Hub administrator account (Security Hub API, AWS CLI)	96
Removing a Security Hub administrator account (console)	96
Removing a Security Hub administrator account (Security Hub API, AWS CLI)	97
Removing the delegated administrator account (Organizations API, AWS CLI)	98
Managing organization member accounts	98
Enabling new accounts automatically	99
Enabling member accounts	100
Disassociating member accounts	101
Managing member accounts by invitation	102
Adding and inviting member accounts	102
Responding to an invitation	104
Disassociating member accounts	106

Deleting member accounts	107
Disassociating from your administrator account	107
Effect of account actions on Security Hub data	108
Security Hub disabled	108
Member account disassociated from administrator account	109
Member account is removed from an organization	109
Account is suspended	109
Account is closed	109
Cross-Region aggregation	111
How cross-Region aggregation works	112
Aggregating new data and replicating updates to data	112
Determining the accounts to aggregate data from	112
Viewing the current configuration	113
Viewing the cross-Region aggregation configuration (console)	113
Viewing the current cross-Region aggregation configuration (Security Hub API, AWS CLI)	113
Enabling cross-Region aggregation	114
Enabling cross-Region aggregation (console)	114
Enabling cross-Region aggregation (Security Hub API, AWS CLI)	114
Updating the configuration	115
Updating the cross-Region aggregation configuration (console)	115
Updating the cross-Region aggregation configuration (Security Hub API, AWS CLI)	116
Stopping cross-Region aggregation	116
Stopping cross-Region aggregation (console)	116
Stopping cross-Region aggregation (Security Hub API, AWS CLI)	117
Findings	118
Creating and updating findings	118
Using BatchImportFindings	119
Using BatchUpdateFindings	122
Viewing a cross-Region finding summary	125
Viewing finding lists and details	126
Filtering and grouping findings (console)	126
Viewing finding details (console)	128
Retrieving finding details (Security Hub API, AWS CLI)	129
Taking action on findings	130
Setting the workflow status for findings	130
Sending findings to a custom action	131
Finding format	132
ASFF syntax	132
ASFF examples	176
Types taxonomy for ASFF	274
Insights	277
Viewing and filtering the list of insights	277
Viewing insight results and findings	277
Viewing and taking action on insight results (console)	278
Viewing insight results (Security Hub API, AWS CLI)	278
Viewing findings for an insight result (console)	279
Managed insights	279
Managing custom insights	286
Creating a custom insight (console)	286
Creating a custom insight (Security Hub API, AWS CLI)	287
Modifying a custom insight (console)	287
Modifying a custom insight (Security Hub API, AWS CLI)	288
Creating a new custom insight from a managed insight (console)	288
Deleting a custom insight (console)	289
Deleting a custom insight (Security Hub API, AWS CLI)	289
Product integrations	290
Managing product integrations	290

Viewing and filtering the list of integrations (console)	290
Viewing information about product integrations (Security Hub API, AWS CLI)	291
Enabling an integration	291
Disabling and enabling the flow of findings from an integration (console)	292
Disabling the flow of findings from an integration (Security Hub API, AWS CLI)	292
Enabling the flow of findings from an integration (Security Hub API, AWS CLI)	292
Viewing the findings from an integration	293
AWS service integrations	293
AWS Audit Manager (Receives findings)	294
AWS Chatbot (Sends findings)	294
AWS Config (Sends findings)	294
Amazon Detective (Linked from Security Hub)	298
AWS Firewall Manager (Sends findings)	298
Amazon GuardDuty (Sends findings)	298
AWS Health (Sends findings)	298
IAM Access Analyzer (Sends findings)	302
Amazon Inspector (Sends findings)	302
Amazon Macie (Sends findings)	303
AWS Systems Manager Explorer and OpsCenter (Receives and updates findings)	303
AWS Systems Manager Patch Manager (Sends findings)	303
AWS Trusted Advisor (Receives findings)	304
Third-party product integrations	304
3CORESec – 3CORESec NTA (Sends findings)	304
Alert Logic – SIEMless Threat Management (Sends findings)	305
Aqua Security – Aqua Cloud Native Security Platform (Sends findings)	305
Aqua Security – Kube-bench (Sends findings)	305
Armor – Armor Anywhere (Sends findings)	305
Atlassian – Jira Service Management (Receives and updates findings)	306
Atlassian – Opsgenie (Receives findings)	306
AttackIQ – AttackIQ (Sends findings)	306
Barracuda Networks – Cloud Security Guardian (Sends findings)	306
BigID – BigID Enterprise (Sends findings)	307
Blue Hexagon – Blue Hexagon forAWS (Sends findings)	307
Capitis Solutions – C2VS (Sends findings)	307
Caveonix – Caveonix Cloud (Sends and receives findings)	307
Check Point – CloudGuard IaaS (Sends findings)	308
Check Point – CloudGuard Posture Management (Sends findings)	308
Cloud Custodian – Cloud Custodian (Sends and receives findings)	308
Cloud Storage Security – Antivirus for Amazon S3 (Sends findings)	308
cloudtamer.io – cloudtamer.io (Sends and receives findings)	309
CrowdStrike – CrowdStrike Falcon (Sends findings)	309
CyberArk – Privileged Threat Analytics (Sends findings)	309
Data Theorem (Sends findings)	310
DisruptOps, Inc. – DisruptOPS (Sends and receives findings)	310
FireEye – FireEye Helix (Receives findings)	310
Forcepoint – Forcepoint CASB (Sends findings)	310
Forcepoint – Forcepoint Cloud Security Gateway (Sends findings)	310
Forcepoint – Forcepoint DLP (Sends findings)	311
Forcepoint – Forcepoint NGFW (Sends findings)	311
Fugue – Fugue (Sends findings)	311
Guardicore – Centra 4.0 (Sends findings)	311
Guardicore – Infection Monkey (Sends findings)	312
HackerOne – Vulnerability Intelligence (Sends findings)	312
Helecloud – Managed Security (Receives findings)	312
IBM – QRadar (Receives findings)	312
Juniper Networks – vSRX Next Generation Firewall (Sends findings)	313
k9 Security – Access Analyzer (Sends findings)	313

Lacework (Sends findings)	313
Logz.io Cloud SIEM (Receives findings)	313
McAfee – MVISION Cloud Native Application Protection Platform (CNAPP) (Sends findings)	314
MicroFocus – MicroFocus Arcsight (Receives findings)	314
NETSCOUT – NETSCOUT Cyber Investigator (Sends findings)	314
PagerDuty – PagerDuty (Receives findings)	314
Palo Alto Networks – Cortex XSOAR (Receives findings)	315
Palo Alto Networks – Prisma Cloud Compute (Sends findings)	315
Palo Alto Networks – Prisma Cloud Enterprise (Sends findings)	315
Palo Alto Networks – VM-Series (Receives findings)	315
Prowler (Sends findings)	316
Qualys – Vulnerability Management (Sends findings)	316
Rackspace Technology – Cloud Native Security (Receives findings)	316
Rapid7 – InsightConnect (Receives findings)	316
Rapid7 – InsightVM (Sends findings)	316
RSA – RSA Archer (Receives findings)	317
SecureCloudDB – SecureCloudDB (Sends findings)	317
SentinelOne (Sends findings)	317
ServiceNow – ITSM (Receives and updates findings)	317
Slack – Slack (Receives findings)	318
Sonrai Security – Sonrai Dig (Sends findings)	318
Sophos – Server Protection (Sends findings)	318
Splunk – Splunk Enterprise (Receives findings)	318
Splunk – Splunk Phantom (Receives findings)	319
StackRox – StackRox Kubernetes Security (Sends findings)	319
Sumo Logic – Machine Data Analytics (Sends findings)	319
Symantec – Cloud Workload Protection (Sends findings)	319
Sysdig – Sysdig Secure for cloud (Sends findings)	320
Tenable – Tenable.io (Sends findings)	320
ThreatModeler (Receives findings)	320
Turbot – Turbot (Sends and receives findings)	320
Vectra AI – Cognito Detect (Sends findings)	320
Using custom product integrations	321
Requirements and recommendations for sending findings from custom security products	321
Updating findings from custom products	322
Example custom integrations	322
Standards and controls	323
Running security checks	323
How Security Hub uses AWS Config rules to run security checks	324
Schedule for running security checks	324
Generating and updating control findings	325
Determining the control status	331
Determining the standard security score	332
Viewing and managing standards	333
Disabling or enabling a security standard	333
Viewing details for a standard	336
Viewing and managing controls	339
Viewing details for a control	339
Enabling new controls automatically	340
Disabling and enabling individual controls	341
Viewing and taking action on control findings	343
Available standards	346
CIS AWS Foundations Benchmark	346
PCI DSS	398
AWS Foundational Security Best Practices standard	457
Security Hub with CloudTrail	602
Security Hub information in CloudTrail	602

Example: Security Hub log file entries	603
Automated response and remediation	604
Types of EventBridge integration	604
All findings (Security Hub Findings - Imported)	605
Findings for custom actions (Security Hub Findings - Custom Action)	605
Insight results for custom actions (Security Hub Insight Results)	605
EventBridge event formats	606
Security Hub Findings - Imported	606
Security Hub Findings - Custom Action	606
Security Hub Insight Results	607
Configuring a rule for automatically sent findings	608
Format of the event pattern	608
Creating an event rule	609
.....	612
Configuring and using custom actions	612
Creating a custom action (console)	612
Creating a custom action (Security Hub API, AWS CLI)	612
Defining a rule in EventBridge	613
Selecting a custom action for findings and insight results	614
Disabling AWS Security Hub	616
Disabling Security Hub (console)	616
Disabling Security Hub (Security Hub API, AWS CLI)	616
Document history	618

What is AWS Security Hub?

AWS Security Hub provides you with a comprehensive view of your security state in AWS and helps you check your environment against security industry standards and best practices.

Security Hub collects security data from across AWS accounts, services, and supported third-party partner products and helps you analyze your security trends and identify the highest priority security issues.

Topics

- [Benefits of AWS Security Hub \(p. 1\)](#)
- [How Security Hub works \(p. 1\)](#)
- [AWS Security Hub free trial, usage, and pricing \(p. 2\)](#)

Benefits of AWS Security Hub

Reduced effort to collect and prioritize findings

Security Hub reduces the effort to collect and prioritize security findings across accounts from integrated AWS services and AWS partner products. Security Hub processes finding data using a standard finding format, which eliminates the need to manage findings data from multiple formats. Security Hub then correlates findings across providers to help you prioritize the most important ones.

Automatic security checks against best practices and standards

Security Hub automatically runs continuous, account-level configuration and security checks based on AWS best practices and industry standards. Security Hub provides the result of these checks as a readiness score, and identifies specific accounts and resources that require attention.

Consolidated view of findings across accounts and providers

Security Hub consolidates your security findings across accounts and provider products and displays results on the Security Hub console. This allows you to view your overall current security status to spot trends, identify potential issues, and take the necessary remediation steps.

Ability to automate remediation of findings

Security Hub supports integration with Amazon EventBridge. To automate remediation of specific findings, you can define custom actions to take when a finding is received. For example, you can configure custom actions to send findings to a ticketing system or to an automated remediation system.

How Security Hub works

You can use Security Hub in the following ways:

Security Hub console

Sign in to the AWS Management Console and open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.

Security Hub API

To access Security Hub programmatically, use the Security Hub API, which allows you to issue HTTPS requests directly to the service. For more information, see the [AWS Security Hub API Reference](#).

When you enable Security Hub, it begins to consume, aggregate, organize, and prioritize findings from AWS services that you have enabled, such as [Amazon GuardDuty](#), [Amazon Inspector](#), and [Amazon Macie](#). You can also enable integrations with AWS partner security products. Those partner products can then also send findings to Security Hub. See [Product integrations](#) (p. 290).

Security Hub also generates its own findings by running continuous, automated security checks based on AWS best practices and supported industry standards. See [Standards and controls](#) (p. 323).

Security Hub then correlates and consolidates findings across providers to help you to prioritize the most significant findings. See [the section called "Viewing finding lists and details"](#) (p. 126) and [the section called "Taking action on findings"](#) (p. 130).

You can also create *insights* in Security Hub. An insight is a collection of findings that are grouped together when you apply a **Group by** filter. Insights help you identify common security issues that may require remediation action. Security Hub includes several managed insights, or you can create your own custom insights. See [Insights](#) (p. 277).

Important

Security Hub only detects and consolidates findings that are generated after you enable Security Hub. It does not retroactively detect and consolidate security findings that were generated before you enabled Security Hub.

Security Hub only receives and processes findings from the Region where you enabled Security Hub in your account.

For full compliance with CIS AWS Foundations Benchmark security checks, you must enable Security Hub in all AWS Regions.

AWS Security Hub free trial, usage, and pricing

When you enable Security Hub for the first time, your AWS account is automatically enrolled in a 30-day Security Hub free trial.

When you use Security Hub during the free trial, you are charged for usage of other services that Security Hub interacts with, such as AWS Config items. You are not charged for AWS Config rules that are enabled by Security Hub security standards.

You are not charged for using Security Hub until your free trial ends.

Note

The Security Hub free trial is not supported in the China (Beijing) Region.

Viewing usage details and estimated cost

Security Hub provides usage information, including an estimated 30-day cost for using Security Hub. The usage details include the time remaining for the free trial. During the free trial, the usage information can help you to understand what the Security Hub cost will be after the free trial ends.

To display the usage information

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.

2. In the navigation pane, choose **Settings**.
3. On the **Settings** page, choose **Usage**.

The estimated monthly cost is based on your account's Security Hub usage for findings and security checks projected over a 30-day period.

The usage information and estimated cost are only for the current Region, not for all Regions in which Security Hub is enabled. In an aggregation Region, the usage information and estimated cost do not include linked Regions. For more information about linked Regions, see [the section called "How cross-Region aggregation works"](#) (p. 112).

Viewing pricing details

For more information about how Security Hub charges for ingested findings and security checks, see [Security Hub pricing](#).

Terminology and concepts

This topic describes the key concepts in AWS Security Hub to help you get started.

Account

A standard Amazon Web Services (AWS) account that contains your AWS resources. You can sign in to AWS with your account and enable Security Hub.

An account can invite other accounts to enable Security Hub and become associated with that account in Security Hub. Accepting a membership invitation is optional. If the invitations are accepted, the account becomes an administrator account, and the added accounts are member accounts. Administrator accounts can view findings in their member accounts.

If you are enrolled in AWS Organizations, then your organization designates a Security Hub administrator account for the organization. The Security Hub administrator account can enable other organization accounts as member accounts.

An account cannot be both an administrator account and a member account at the same time. An account can only have one administrator account.

For more information, see [Managing administrator and member accounts \(p. 89\)](#).

Administrator account

An account in Security Hub that is granted access to view findings for associated member accounts.

An account becomes an administrator account in one of the following ways:

- The account invites other accounts to become associated with it in Security Hub. When those accounts accept the invitation, they become member accounts, and the inviting account becomes their administrator account.
- The account is designated by an organization management account as the Security Hub administrator account. The Security Hub administrator account can enable any organization account as a member account, and can also invite other accounts to be member accounts.

An account can only have one administrator account. An account cannot be both an administrator account and a member account at the same time.

Aggregation Region

Setting an aggregation Region allows you to view security findings from multiple Regions in a single pane of glass.

The aggregation Region is the Region from which you view and manage findings. Findings are aggregated to the aggregation Region from linked Regions. Updates to findings are replicated across Regions.

In the aggregation Region, the **Security standards**, **Insights**, and **Findings** pages include data from all linked Regions.

See [Cross-Region aggregation \(p. 111\)](#).

Archived finding

A finding that has a `RecordState` set to `ARCHIVED`. Archiving a finding indicates that the finding provider believes that the finding is no longer relevant. The record state is separate from the workflow status, which tracks the status of an investigation into a finding.

Finding providers can use the [BatchImportFindings](#) operation of the Security Hub API to archive findings that they created. Security Hub automatically archives findings for controls if the control is disabled or the associated resource is deleted, based on one of the following criteria.

- The finding is not updated in three to five days (note that this is best effort and not guaranteed).
- The associated AWS Config evaluation returns `NOT_APPLICABLE`.

By default, archived findings are excluded from findings lists in the Security Hub console. You can update the filter to include archived findings.

The `GetFindings` operation of the Security Hub API returns both active and archived findings. You can include a filter for the record state.

```
"RecordState": [  
  {  
    "Comparison": "EQUALS",  
    "Value": "ARCHIVED"  
  }  
],
```

AWS Security Finding Format (ASFF)

A standardized format for the contents of findings that Security Hub aggregates or generates. The AWS Security Finding Format enables you to use Security Hub to view and analyze findings that are generated by AWS security services, third-party solutions, or Security Hub itself from running security checks. For more information, see [AWS Security Finding Format \(ASFF\) \(p. 132\)](#).

Control

A safeguard or countermeasure prescribed for an information system or an organization designed to protect the confidentiality, integrity, and availability of its information and to meet a set of defined security requirements. A security standard consists of controls.

Custom action

A Security Hub mechanism for sending selected findings to EventBridge. A custom action is created in Security Hub. It is then linked to an EventBridge rule. The rule defines a specific action to take when a finding is received that is associated with the custom action ID. Custom actions can be used, for example, to send a specific finding, or a small set of findings, to a response or remediation workflow. For more information, see [the section called "Creating a custom action \(console\)" \(p. 612\)](#).

Delegated administrator account (Organizations)

In Organizations, the delegated administrator account for a service is able to manage the use of a service for the organization.

In Security Hub, the Security Hub administrator account is also the delegated administrator account for Security Hub. When the organization management account first designates a Security Hub administrator account, Security Hub calls Organizations to make that account the delegated administrator account.

The organization management account must then choose the delegated administrator account as the Security Hub administrator account in all Regions.

Finding

The observable record of a security check or security-related detection.

For more information about findings in Security Hub, see [Findings \(p. 118\)](#).

Note

Findings are deleted 90 days after the most recent update or 90 days after the creation date if no update occurs. To store findings for longer than 90 days, you can configure a rule in EventBridge that routes findings to your Amazon S3 bucket.

Cross-Region aggregation

The aggregation of findings, insights, control compliance statuses, and security scores from linked Regions to an aggregation Region. You can then view all of your data from the aggregation Region and update findings and insights from the aggregation Region.

See [Cross-Region aggregation \(p. 111\)](#).

Finding ingestion

The import of findings into Security Hub from other AWS services and from third-party partner providers.

Finding ingestion events include both new findings and updates to existing findings.

Insight

A collection of related findings defined by an aggregation statement and optional filters. An insight identifies a security area that requires attention and intervention. Security Hub offers several managed (default) insights that you can't modify. You can also create custom Security Hub insights to track security issues that are unique to your AWS environment and usage. For more information, see [Insights \(p. 277\)](#).

Linked Region

When you enable cross-Region aggregation, a linked Region is a region that aggregates findings, insights, control compliance statuses, and security scores to the aggregation Region.

In a linked Region, the **Findings** and **Insights** pages contain findings only from that Region.

See [Cross-Region aggregation \(p. 111\)](#).

Member account

An account that has granted permission to an administrator account to view and take action on their findings.

An account becomes a member account in one of the following ways:

- The account accepts an invitation from another account.
- For an organization account, the Security Hub administrator account enables the account as a member account.

Related requirements

A set of industry or regulatory requirements that are mapped to a control.

Rule

A set of automated criteria that is used to assess whether a control is being adhered to. When a rule is evaluated, it can pass or fail. If the evaluation cannot determine whether rule passes or fails, then the rule is in a warning state. If the rule cannot be evaluated, then it is in a not available state.

Security check

A specific point-in-time evaluation of a rule against a single resource resulting in a passed, failed, warning, or not available state. Running a security check produces a finding.

Security Hub administrator account

An organization account that manages Security Hub membership for an organization.

The organization management account designates the Security Hub administrator account in each Region. The organization management account must choose the same Security Hub administrator account in all Regions.

The Security Hub administrator account is also the delegated administrator account for Security Hub in Organizations.

The Security Hub administrator account can enable any organization account as a member account. The Security Hub administrator account can also invite other accounts to be member accounts.

Security standard

A published statement on a topic specifying the characteristics, usually measurable and in the form of controls, that must be satisfied or achieved for compliance. Security standards can be based on regulatory frameworks, best practices, or internal company policies. To learn more about security standards in Security Hub, see [Standards and controls \(p. 323\)](#).

Severity

The severity assigned to a Security Hub control identifies the importance of the control. The severity of a control can be **Critical**, **High**, **Medium**, **Low**, or **Informational**. The severity assigned to control findings is equal to the severity of the control itself. To learn about how Security Hub assigns severity to a control, see [Severity \(p. 329\)](#).

Workflow status

The status of an investigation into a finding. Tracked using the `WorkflowStatus` attribute.

The workflow status is initially `NEW`. If you notified the resource owner to take action on the finding, you can set the workflow status to `NOTIFIED`. If the finding is not an issue, and does not require any action, set the workflow status to `SUPPRESSED`. After you review and remediate a finding, set the workflow status to `RESOLVED`.

By default, most finding lists only include findings with a workflow status of `NEW` or `NOTIFIED`. Finding lists for controls also include `RESOLVED` findings.

For the [GetFindings](#) operation, you can include a filter for the workflow status.

```
"WorkflowStatus": [  
  {  
    "Comparison": "EQUALS",  
    "Value": "RESOLVED"  
  }  
],
```

The Security Hub console provides an option to set the workflow status for findings. Customers (or SIEM, ticketing, incident management, or SOAR tools working on behalf of a customer to update findings from finding providers) can also use [BatchUpdateFindings](#) to update the workflow status.

Prerequisites and recommendations

You must have an AWS account to enable AWS Security Hub. If you don't have an account, use the following procedure to create one.

To sign up for AWS

1. Open <https://portal.aws.amazon.com/billing/signup>.
2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code on the phone keypad.

Security Hub strongly recommends that you enable AWS Organizations. Using Organizations to manage your accounts streamlines the process of managing member accounts.

Security Hub requires that AWS Config is enabled in all accounts that have Security Hub enabled. Security Hub controls use AWS Config rules to complete security checks.

Topics

- [Using AWS Organizations to manage accounts \(p. 8\)](#)
- [Enabling and configuring AWS Config \(p. 8\)](#)

Using AWS Organizations to manage accounts

To help automate and streamline management of accounts, Security Hub strongly recommends that you enable AWS Organizations.

If you have Organizations enabled, then Security Hub automatically detects new accounts as they are added to your organization.

For information on setting up Organizations, see [Creating and managing an organization](#) in the *AWS Organizations User Guide*.

After you set up your organization, your organization management account designates the Security Hub administrator account. See [the section called "Designating a Security Hub administrator account" \(p. 93\)](#). The Security Hub administrator account is also the delegated administrator account in Organizations.

The Security Hub administrator account then enables and manages other organization accounts as Security Hub member accounts. See [the section called "Managing organization member accounts" \(p. 98\)](#).

Enabling and configuring AWS Config

AWS Security Hub uses service-linked AWS Config rules to perform most of its security checks for controls.

To support these controls, AWS Config must be enabled on all accounts – both the administrator account and member accounts – in each Region where Security Hub is enabled. AWS Config must be configured to record at a minimum the resources that are required for the standards that you have enabled.

- [the section called “Required AWS Config resources” \(p. 347\)](#)
- [the section called “Required AWS Config resources” \(p. 399\)](#)
- [the section called “Required AWS Config resources” \(p. 457\)](#)

Security Hub recommends that you enable resource recording in AWS Config before you enable Security Hub standards. If Security Hub tries to run security checks when resource recording is not enabled, the checks return errors.

Security Hub does not manage AWS Config for you. If you already have AWS Config enabled, you can continue to configure its settings through the AWS Config console or APIs.

If you enable AWS Config after you enable a standard, Security Hub still creates the AWS Config rules, but only if you enable AWS Config within 31 days after you enable the standard. If you do not enable AWS Config within 31 days, then you must disable and re-enable the standard after you enable AWS Config.

After you enable a standard, Security Hub tries to create the AWS Config rules up to six times during the 31 days.

- On the day you enable the standard
- The day after you enable the standard
- Three days after you enable the standard
- Seven days after you enable the standard
- 15 days after you enable the standard
- 31 days after you enable the standard

How to enable AWS Config

If you have not set up AWS Config already, you can set it up in one of the following ways:

1. **Console or CLI** – You can manually enable AWS Config using the AWS Config console or CLI. See [Getting started with AWS Config](#) in the *AWS Config Developer Guide*.
2. **AWS CloudFormation template** – If you have integrated with AWS Organizations or want to enable AWS Config on a large set of accounts, you can easily enable AWS Config with the CloudFormation template **Enable AWS Config**. To access this template, see [AWS CloudFormation StackSets sample templates](#) in the *AWS CloudFormation User Guide*. For more details about using this template, see [Managing AWS Organizations accounts using AWS Config and AWS CloudFormation StackSets](#).
3. **Github script** – Security Hub offers a [script in GitHub](#) that allows you to enable multiple accounts across Regions. This script is useful if you have not integrated with Organizations or if you have accounts that are not part of your organization. When you use this script to enable Security Hub, it also automatically enables AWS Config for these accounts.

Configuring resource recording in AWS Config

When you enable resource recording during AWS Config setup, AWS Config records all supported types of *regional resources* that AWS Config discovers in the region in which it is running. You can also configure AWS Config to record supported types of *global resources*. You only need to record global resources in a single Region.

If you are using CloudFormation StackSets to enable AWS Config, we recommend that you run two different StackSets. Run one StackSet to record all resources, including global resources, in a single Region. Run a second StackSet to record all resources except global resources in other Regions.

You can also use Quick Setup, a capability of AWS Systems Manager, to quickly configure resource recording in AWS Config across your accounts and Regions. During Quick Setup, you can choose which Region you would like to record global resources in. For more information, see [AWS Config recording](#) in the *AWS Systems Manager User Guide*.

If you do not record global resources in all Regions, then in the Regions where you do not record global resources, you must disable the control [the section called "2.5 – Ensure AWS Config is enabled"](#) (p. 363). CIS 2.5 generates failed findings in Regions where global resources are not recorded. For details about other controls that you may want to disable in Regions where global resources are not recorded, see the following topics.

- [the section called "Controls that you might want to disable"](#) (p. 397)
- [the section called "Controls that you might want to disable"](#) (p. 456)
- [the section called "Controls that you might want to disable"](#) (p. 601)

Note that if you use the [multi-account script](#) to enable Security Hub, it automatically enables resource recording for all resources, including global resources, in all Regions. It does not limit recording of global resources to a single Region. You can then update the configuration to only record global resources in a single Region. See [Selecting which resources AWS Config records](#) in the *AWS Config Developer Guide*.

The following topics list the required resources for each standard. You can enable recording only for the required resources. However, Security Hub continues to add new controls and support new resources.

- [the section called "Required AWS Config resources"](#) (p. 347)
- [the section called "Required AWS Config resources"](#) (p. 399)
- [the section called "Required AWS Config resources"](#) (p. 457)

For details about the costs associated with resource recording, see the [AWS Config pricing page](#).

AWS Security Hub quotas

Maximum quotas

The following Security Hub quotas are per AWS account per Region.

Resource	Quota	Comments
Number of Security Hub member accounts	5,000	<p>The maximum number of Security Hub member accounts that can be added per administrator account per Region.</p> <p>This is a hard quota. You cannot request an increase to the allowed number of Security Hub member accounts.</p>
Number of Security Hub outstanding invitations	1,000	<p>The maximum number of outstanding Security Hub member account invitations that can be sent per administrator account per Region.</p> <p>This is a hard quota. You cannot request an increase to the allowed number of Security Hub outstanding invitations.</p>
Number of custom actions	50	<p>The maximum number of Security Hub custom actions that can be created.</p> <p>This is a hard quota. You cannot request an increase to the number of custom actions.</p>
Number of custom insights	100	<p>The maximum number of user-defined custom insights that can be created.</p> <p>This is a hard quota. You cannot request an increase to the allowed number of Security Hub custom insights.</p>
Number of insight results	100	<p>The maximum number of aggregated results returned for the <code>GetInsightsResults</code> API operation.</p> <p>This is a hard quota. You cannot request an increase to the number of insight results.</p>
Number of service-linked AWS Config rules	250	<p>The maximum number of service-linked AWS Config rules that Security Hub creates to perform security checks for controls.</p> <p>This is a hard quota. You cannot request an increase to the number of service-linked AWS Config rules.</p>
Security Hub finding retention time	90 days	<p>Findings are deleted 90 days after the most recent update or 90 days after the creation date if no update occurs.</p> <p>To store findings for longer than 90 days, you can configure a rule in EventBridge that routes findings to your Amazon S3 bucket.</p>

Rate quotas

The following AWS Security Hub quotas are per AWS account per Region.

Request type	Rate limit quota (per second)	Burst limit quota (per second)
<code>BatchEnableStandards</code>	1	1
<code>GetFindings</code>	3	6
<code>BatchImportFindings</code>	10	30
<code>BatchUpdateFindings</code>	10	30
<code>UpdateStandardsControl</code>	1	5
All other request types	10	30

If you have set up [Cross-Region aggregation \(p. 111\)](#), one call to `BatchImportFindings` and `BatchUpdateFindings` impacts linked Regions and the aggregation Region. The `GetFindings` operation retrieves findings from linked Regions and the aggregation Region. However, the `BatchEnableStandards` and `UpdateStandardsControl` operations are Region-specific.

Regional limits

Some AWS Security Hub features are available in only some AWS Regions. The following sections specify these Regional limits.

Contents

- [Cross-Region aggregation restrictions \(p. 13\)](#)
- [Integrations not supported in all Regions \(p. 14\)](#)
 - [Integrations that are supported in China \(Beijing\) and China \(Ningxia\) \(p. 14\)](#)
 - [Integrations that are supported in AWS GovCloud \(US-East\) and AWS GovCloud \(US-West\) \(p. 14\)](#)
- [Controls not supported in all Regions \(p. 15\)](#)
 - [US East \(Ohio\) \(p. 15\)](#)
 - [US West \(N. California\) \(p. 16\)](#)
 - [US West \(Oregon\) \(p. 17\)](#)
 - [Africa \(Cape Town\) \(p. 18\)](#)
 - [Asia Pacific \(Hong Kong\) \(p. 21\)](#)
 - [Asia Pacific \(Jakarta\) \(p. 22\)](#)
 - [Asia Pacific \(Mumbai\) \(p. 29\)](#)
 - [Asia Pacific \(Osaka\) \(p. 30\)](#)
 - [Asia Pacific \(Seoul\) \(p. 34\)](#)
 - [Asia Pacific \(Singapore\) \(p. 34\)](#)
 - [Asia Pacific \(Sydney\) \(p. 35\)](#)
 - [Asia Pacific \(Tokyo\) \(p. 36\)](#)
 - [Canada \(Central\) \(p. 37\)](#)
 - [China \(Beijing\) \(p. 37\)](#)
 - [China \(Ningxia\) \(p. 42\)](#)
 - [Europe \(Frankfurt\) \(p. 47\)](#)
 - [Europe \(Ireland\) \(p. 47\)](#)
 - [Europe \(London\) \(p. 48\)](#)
 - [Europe \(Milan\) \(p. 49\)](#)
 - [Europe \(Paris\) \(p. 52\)](#)
 - [Europe \(Stockholm\) \(p. 53\)](#)
 - [Middle East \(Bahrain\) \(p. 54\)](#)
 - [South America \(São Paulo\) \(p. 55\)](#)
 - [AWS GovCloud \(US-East\) \(p. 56\)](#)
 - [AWS GovCloud \(US-West\) \(p. 61\)](#)

Cross-Region aggregation restrictions

[Cross-Region aggregation \(p. 111\)](#) is not available in the AWS GovCloud (US) Region. In the China Regions, cross-Region aggregation is only available across the China Regions (specifically, you can only aggregate data from China (Beijing) to China (Ningxia), or the other way around).

You can't use a Region that is disabled by default as your aggregation Region. For a list of Regions that are disabled by default, see [Enabling a Region](#) in the *AWS General Reference*.

Integrations not supported in all Regions

Some integrations are not available in all Regions. If an integration is not available in a specific Region, it is not listed on the **Integrations** page of the Security Hub console when you choose that Region.

Integrations that are supported in China (Beijing) and China (Ningxia)

The China (Beijing) and China (Ningxia) Regions only support the following [integrations with AWS services](#) (p. 293):

- AWS Firewall Manager
- Amazon GuardDuty
- IAM Access Analyzer
- Systems Manager Explorer
- Systems Manager OpsCenter
- Systems Manager Patch Manager

The China (Beijing) and China (Ningxia) Regions only support the following [third-party integrations](#) (p. 304):

- Cloud Custodian
- FireEye Helix
- Helecloud
- IBM QRadar
- PagerDuty
- Palo Alto Networks Cortex XSOAR
- Palo Alto Networks VM-Series
- Prowler
- RSA Archer
- Splunk Enterprise
- Splunk Phantom
- ThreatModeler

Integrations that are supported in AWS GovCloud (US-East) and AWS GovCloud (US-West)

The AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions only support the following [integrations with AWS services](#) (p. 293):

- Amazon Detective
- AWS Firewall Manager
- Amazon GuardDuty

- Amazon Inspector
- IAM Access Analyzer

The AWS GovCloud (US-East) and AWS GovCloud (US-West) Regions only support the following [third-party integrations \(p. 304\)](#):

- Atlassian Jira Service Manager
- Atlassian OpsGenie
- Caveonix Cloud
- Cloud Custodian
- Cloud Storage Security Antivirus for Amazon S3
- cloudtamer.io
- CrowdStrike Falcon
- FireEye Helix
- Forcepoint CASB
- Forcepoint DLP
- Forcepoint NGFW
- MicroFocus ArcSight
- NETSCOUT Cyber Investigator
- PagerDuty
- Palo Alto Networks – Prisma Cloud Compute
- Palo Alto Networks – Prisma Cloud Enterprise
- Palo Alto Networks – VM-Series
- Prowler
- Rackspace Technology – Cloud Native Security
- Rapid7 InsightConnect
- RSA Archer
- SecureCloudDb
- ServiceNow ITSM
- Slack
- ThreatModeler
- Vectra AI Cognito Detect

Controls not supported in all Regions

The following Regions do not support all of the Security Hub controls. For each Region, this list shows the controls that are not supported.

US East (Ohio)

The following controls are not supported in US East (Ohio).

[the section called “AWS Foundational Security Best Practices standard” \(p. 457\)](#)

[the section called “\[CloudFront.1\] CloudFront distributions should have a default root object configured” \(p. 470\)](#)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

US West (N. California)

The following controls are not supported in US West (N. California).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

US West (Oregon)

The following controls are not supported in US West (Oregon).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Africa (Cape Town)

The following controls are not supported in Africa (Cape Town).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.4 – Ensure access keys are rotated every 90 days or less ” (p. 351)

the section called “1.20 - Ensure a support role has been created to manage incidents with AWS Support ” (p. 358)

the section called “4.1 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 22 ” (p. 395)

the section called “4.2 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 ” (p. 396)

the section called “PCI DSS” (p. 398)

the section called “[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 405)

the section called “[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 406)

the section called “[PCI.DMS.1] AWS Database Migration Service replication instances should not be public” (p. 411)

the section called “[PCI.EC2.1] Amazon EBS snapshots should not be publicly restorable” (p. 413)

the section called “[PCI.EC2.4] Unused EC2 EIPs should be removed” (p. 416)

the section called “[PCI.EC2.5] Security groups should not allow ingress from 0.0.0.0/0 to port 22” (p. 417)

the section called “[PCI.ELBV2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 419)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.IAM.1] IAM root user access key should not exist” (p. 424)

the section called “[PCI.RDS.1] Amazon RDS snapshots should prohibit public access” (p. 439)

the section called “[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access” (p. 451)

the section called “[PCI.SSM.1] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation” (p. 452)

the section called “[PCI.SSM.2] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 454)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period” (p. 460)

the section called “[APIGateway.1] API Gateway REST and WebSocket API logging should be enabled” (p. 461)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 479)

the section called “[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 480)

the section called “[DMS.1] AWS Database Migration Service replication instances should not be public” (p. 483)

the section called “[EC2.1] Amazon EBS snapshots should not be public, determined by the availability to be restorable by anyone” (p. 485)

the section called “[EC2.3] Attached EBS volumes should be encrypted at rest” (p. 487)

the section called “[EC2.4] Stopped EC2 instances should be removed after a specified time period” (p. 488)

the section called “[EC2.8] EC2 instances should use IMDSv2” (p. 490)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[EFS.1] Amazon EFS should be configured to encrypt file data at rest using AWS KMS” (p. 509)

the section called “[EFS.2] Amazon EFS volumes should be in backup plans” (p. 510)

the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)

the section called “[ELB.4] Application load balancers should be configured to drop HTTP headers” (p. 515)

the section called “[ELB.8] Classic Load Balancers with HTTPS/SSL listeners should use a predefined security policy that has strong configuration” (p. 518)

the section called “[ELBv2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 522)

the section called “[EMR.1] Amazon EMR cluster master nodes should not have public IP addresses” (p. 523)

the section called “[ES.3] Elasticsearch domains should encrypt data sent between nodes” (p. 524)

the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)

the section called “[IAM.3] IAM users' access keys should be rotated every 90 days or less” (p. 530)

the section called “[IAM.4] IAM root user access key should not exist” (p. 531)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[RDS.1] RDS snapshots should be private” (p. 550)

the section called “[RDS.9] Database logging should be enabled” (p. 555)

the section called “[RDS.10] IAM authentication should be configured for RDS instances” (p. 558)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled” (p. 571)

the section called “[S3.1] S3 Block Public Access setting should be enabled” (p. 574)

the section called “[SageMaker.1] SageMaker notebook instances should not have direct internet access” (p. 584)

the section called “[SSM.2] All EC2 instances managed by Systems Manager should be compliant with patching requirements” (p. 591)

the section called “[SSM.3] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 592)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Hong Kong)

The following controls are not supported in Asia Pacific (Hong Kong).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[RDS.10] IAM authentication should be configured for RDS instances” (p. 558)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Jakarta)

The following controls are not supported in Asia Pacific (Jakarta).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.12 – Ensure no root user access key exists” (p. 354)

the section called “1.20 – Ensure a support role has been created to manage incidents with AWS Support ” (p. 358)

the section called “2.9 – Ensure VPC flow logging is enabled in all VPCs ” (p. 366)

the section called “4.1 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 22 ” (p. 395)

the section called “4.2 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 ” (p. 396)

the section called “4.3 – Ensure the default security group of every VPC restricts all traffic” (p. 397)

the section called “PCI DSS” (p. 398)

the section called “[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 405)

the section called “[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 406)

the section called “[PCI.DMS.1] AWS Database Migration Service replication instances should not be public” (p. 411)

the section called “[PCI.EC2.1] Amazon EBS snapshots should not be publicly restorable” (p. 413)

the section called “[PCI.EC2.2] VPC default security group should prohibit inbound and outbound traffic” (p. 414)

the section called “[PCI.EC2.5] Security groups should not allow ingress from 0.0.0.0/0 to port 22” (p. 417)

the section called “[PCI.EC2.6] VPC flow logging should be enabled in all VPCs” (p. 418)

the section called “[PCI.ELBV2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 419)

the section called “[PCI.ES.1] Elasticsearch domains should be in a VPC” (p. 420)

the section called “[PCI.ES.2] Elasticsearch domains should have encryption at rest enabled” (p. 422)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.IAM.1] IAM root user access key should not exist” (p. 424)

the section called “[PCI.Lambda.2] Lambda functions should be in a VPC” (p. 435)

the section called “[PCI.OpenSearch.1] Amazon OpenSearch Service domains should be in a VPC” (p. 437)

the section called “[PCI.OpenSearch.2] OpenSearch domains should have encryption at rest enabled” (p. 438)

the section called “[PCI.RDS.1] Amazon RDS snapshots should prohibit public access” (p. 439)

the section called “[PCI.Redshift.1] Amazon Redshift clusters should prohibit public access” (p. 442)

the section called “[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access” (p. 451)

the section called “[PCI.SSM.1] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation” (p. 452)

the section called “[PCI.SSM.2] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 454)

the section called “[PCI.SSM.3] EC2 instances should be managed by AWS Systems Manager” (p. 455)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[APIGateway.1] API Gateway REST and WebSocket API logging should be enabled” (p. 461)

the section called “[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication” (p. 462)

the section called “[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled” (p. 462)

the section called “[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL” (p. 463)

the section called “[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones” (p. 465)

the section called “[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)” (p. 465)

the section called “[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1” (p. 466)

the section called “[AutoScaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses” (p. 467)

the section called “[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones” (p. 468)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 479)

the section called “[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 480)

the section called “[CodeBuild.4] CodeBuild project environments should have a logging configuration” (p. 481)

the section called “[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled” (p. 481)

the section called “[DMS.1] AWS Database Migration Service replication instances should not be public” (p. 483)

the section called “[DynamoDB.2] DynamoDB tables should have point-in-time recovery enabled” (p. 484)

the section called “[DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest” (p. 485)

the section called “[EC2.1] Amazon EBS snapshots should not be public, determined by the availability to be restorable by anyone” (p. 485)

the section called “[EC2.2] The VPC default security group should not allow inbound and outbound traffic ” (p. 486)

the section called “[EC2.3] Attached EBS volumes should be encrypted at rest” (p. 487)

the section called “[EC2.4] Stopped EC2 instances should be removed after a specified time period” (p. 488)

the section called “[EC2.6] VPC flow logging should be enabled in all VPCs ” (p. 489)

the section called “[EC2.7] EBS default encryption should be enabled” (p. 490)

- the section called “[EC2.8] EC2 instances should use IMDSv2” (p. 490)
- the section called “[EC2.9] EC2 instances should not have a public IP address” (p. 491)
- the section called “[EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service” (p. 492)
- the section called “[EC2.15] EC2 subnets should not automatically assign public IP addresses” (p. 494)
- the section called “[EC2.16] Unused network access control lists should be removed” (p. 495)
- the section called “[EC2.17] EC2 instances should not use multiple ENIs” (p. 495)
- the section called “[EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports” (p. 496)
- the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)
- the section called “[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389” (p. 498)
- the section called “[EC2.22] Unused EC2 security groups should be removed” (p. 499)
- the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)
- the section called “[EC2.27] Running EC2 Instances should not use key pairs” (p. 501)
- the section called “[ECR.1] ECR private repositories should have image scanning configured” (p. 502)
- the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)
- the section called “[ECR.3] ECR repositories should have at least one lifecycle policy configured” (p. 503)
- the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)
- the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)
- the section called “[ECS.5] ECS containers should be limited to read-only access to root filesystems” (p. 506)
- the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)
- the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)
- the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)
- the section called “[EFS.1] Amazon EFS should be configured to encrypt file data at rest using AWS KMS” (p. 509)
- the section called “[EFS.2] Amazon EFS volumes should be in backup plans” (p. 510)
- the section called “[EFS.3] EFS access points should enforce a root directory” (p. 510)

the section called “[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled” (p. 512)

the section called “[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled” (p. 513)

the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)

the section called “[ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination” (p. 514)

the section called “[ELB.4] Application load balancers should be configured to drop HTTP headers” (p. 515)

the section called “[ELB.6] Application Load Balancer deletion protection should be enabled” (p. 517)

the section called “[ELB.8] Classic Load Balancers with HTTPS/SSL listeners should use a predefined security policy that has strong configuration” (p. 518)

the section called “[ELB.9] Classic Load Balancers should have cross-zone load balancing enabled” (p. 518)

the section called “[ELB.10] Classic Load Balancers should span multiple Availability Zones” (p. 519)

the section called “[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 520)

the section called “[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones” (p. 520)

the section called “[ELBv2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 522)

the section called “[EMR.1] Amazon EMR cluster master nodes should not have public IP addresses” (p. 523)

the section called “[ES.1] Elasticsearch domains should have encryption at rest enabled” (p. 523)

the section called “[ES.2] Elasticsearch domains should be in a VPC” (p. 524)

the section called “[ES.3] Elasticsearch domains should encrypt data sent between nodes” (p. 524)

the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)

the section called “[IAM.4] IAM root user access key should not exist” (p. 531)

the section called “[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services” (p. 536)

the section called “[KMS.1] IAM customer managed policies should not allow decryption and re-encryption actions on all KMS keys” (p. 537)

the section called “[KMS.2] IAM principals should not have IAM inline policies that allow decryption and re-encryption actions on all KMS keys ” (p. 538)

the section called “[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone” (p. 542)

the section called “[Network Firewall.3] Network Firewall policies should have at least one rule group associated” (p. 543)

the section called “[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets” (p. 543)

the section called “[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets” (p. 544)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[RDS.1] RDS snapshots should be private” (p. 550)

the section called “[RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest” (p. 552)

the section called “[RDS.6] Enhanced monitoring should be configured for RDS DB instances and clusters” (p. 553)

the section called “[RDS.7] RDS clusters should have deletion protection enabled” (p. 554)

the section called “[RDS.8] RDS DB instances should have deletion protection enabled” (p. 555)

the section called “[RDS.9] Database logging should be enabled” (p. 555)

the section called “[RDS.10] IAM authentication should be configured for RDS instances” (p. 558)

the section called “[RDS.11] Amazon RDS instances should have automatic backups enabled” (p. 559)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.13] RDS automatic minor version upgrades should be enabled” (p. 560)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.16] RDS DB clusters should be configured to copy tags to snapshots” (p. 563)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[RDS.25] RDS database instances should use a custom administrator username” (p. 569)

the section called “[Redshift.1] Amazon Redshift clusters should prohibit public access” (p. 569)

the section called “[Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit” (p. 570)

the section called “[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled” (p. 571)

the section called “[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing” (p. 573)

the section called “[Redshift.8] Amazon Redshift clusters should not use the default Admin username” (p. 573)

the section called “[Redshift.9] Redshift clusters should not use the default database name” (p. 574)

the section called “[S3.8] S3 Block Public Access setting should be enabled at the bucket level” (p. 580)

the section called “[S3.10] S3 buckets with versioning enabled should have lifecycle policies configured” (p. 581)

the section called “[S3.11] S3 buckets should have event notifications enabled” (p. 582)

the section called “[S3.12] S3 access control lists (ACLs) should not be used to manage user access to buckets” (p. 582)

the section called “[S3.13] S3 buckets should have lifecycle policies configured” (p. 583)

the section called “[SageMaker.1] SageMaker notebook instances should not have direct internet access” (p. 584)

the section called “[SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled” (p. 585)

the section called “[SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully” (p. 586)

the section called “[SecretsManager.3] Remove unused Secrets Manager secrets” (p. 586)

the section called “[SNS.1] SNS topics should be encrypted at rest using AWS KMS” (p. 588)

the section called “[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic” (p. 589)

the section called “[SQS.1] Amazon SQS queues should be encrypted at rest” (p. 590)

the section called “[SSM.1] EC2 instances should be managed by AWS Systems Manager” (p. 590)

the section called “[SSM.2] All EC2 instances managed by Systems Manager should be compliant with patching requirements” (p. 591)

the section called “[SSM.3] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 592)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.2] A WAF Regional rule should have at least one condition” (p. 595)

the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)

the section called “[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group” (p. 596)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Mumbai)

The following controls are not supported in Asia Pacific (Mumbai).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Osaka)

The following controls are not supported in Asia Pacific (Osaka).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.12 – Ensure no root user access key exists” (p. 354)

the section called “1.20 - Ensure a support role has been created to manage incidents with AWS Support ” (p. 358)

the section called “4.1 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 22 ” (p. 395)

the section called “4.2 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 ” (p. 396)

the section called “PCI DSS” (p. 398)

the section called “[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 405)

the section called “[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 406)

the section called “[PCI.DMS.1] AWS Database Migration Service replication instances should not be public” (p. 411)

the section called “[PCI.EC2.1] Amazon EBS snapshots should not be publicly restorable” (p. 413)

the section called “[PCI.EC2.5] Security groups should not allow ingress from 0.0.0.0/0 to port 22” (p. 417)

the section called “[PCI.ELBV2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 419)

the section called “[PCI.ES.1] Elasticsearch domains should be in a VPC” (p. 420)

the section called “[PCI.ES.2] Elasticsearch domains should have encryption at rest enabled” (p. 422)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.IAM.1] IAM root user access key should not exist” (p. 424)

the section called “[PCI.Lambda.1] Lambda functions should prohibit public access” (p. 433)

the section called “[PCI.Lambda.2] Lambda functions should be in a VPC” (p. 435)

the section called “[PCI.RDS.1] Amazon RDS snapshots should prohibit public access” (p. 439)

the section called “[PCI.Redshift.1] Amazon Redshift clusters should prohibit public access” (p. 442)

the section called “[PCI.S3.6] S3 Block Public Access setting should be enabled” (p. 449)

the section called “[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access” (p. 451)

the section called “[PCI.SSM.1] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation” (p. 452)

the section called “[PCI.SSM.2] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 454)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication” (p. 462)

the section called “[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled” (p. 462)

the section called “[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL” (p. 463)

the section called “[AutoScaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses” (p. 467)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.4] CodeBuild project environments should have a logging configuration” (p. 481)

the section called “[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled” (p. 481)

the section called “[EC2.15] EC2 subnets should not automatically assign public IP addresses” (p. 494)

the section called “[EC2.16] Unused network access control lists should be removed” (p. 495)

- the section called “[EC2.17] EC2 instances should not use multiple ENIs” (p. 495)
- the section called “[EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports” (p. 496)
- the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)
- the section called “[EC2.22] Unused EC2 security groups should be removed” (p. 499)
- the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)
- the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)
- the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)
- the section called “[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions” (p. 503)
- the section called “[ECS.2] Amazon ECS services should not have public IP addresses assigned to them automatically” (p. 504)
- the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)
- the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)
- the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)
- the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)
- the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)
- the section called “[EFS.4] EFS access points should enforce a user identity” (p. 511)
- the section called “[EKS.2] EKS clusters should run on a supported Kubernetes version” (p. 512)
- the section called “[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled” (p. 512)
- the section called “[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled” (p. 513)
- the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)
- the section called “[ELB.4] Application load balancers should be configured to drop HTTP headers” (p. 515)
- the section called “[ELB.8] Classic Load Balancers with HTTPS/SSL listeners should use a predefined security policy that has strong configuration” (p. 518)
- the section called “[ELB.9] Classic Load Balancers should have cross-zone load balancing enabled” (p. 518)
- the section called “[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services” (p. 536)

- the section called “[KMS.3] AWS KMS keys should not be unintentionally deleted” (p. 539)
- the section called “[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone” (p. 542)
- the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)
- the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)
- the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)
- the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)
- the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)
- the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)
- the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)
- the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)
- the section called “[RDS.9] Database logging should be enabled” (p. 555)
- the section called “[RDS.10] IAM authentication should be configured for RDS instances” (p. 558)
- the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)
- the section called “[RDS.13] RDS automatic minor version upgrades should be enabled” (p. 560)
- the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)
- the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)
- the section called “[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled” (p. 571)
- the section called “[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing” (p. 573)
- the section called “[S3.8] S3 Block Public Access setting should be enabled at the bucket level” (p. 580)
- the section called “[SecretsManager.3] Remove unused Secrets Manager secrets” (p. 586)
- the section called “[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days” (p. 587)
- the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)
- the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)
- the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)
- the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Seoul)

The following controls are not supported in Asia Pacific (Seoul).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Singapore)

The following controls are not supported in Asia Pacific (Singapore).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Sydney)

The following controls are not supported in Asia Pacific (Sydney).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled” (p. 571)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Asia Pacific (Tokyo)

The following controls are not supported in Asia Pacific (Tokyo).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Canada (Central)

The following controls are not supported in Canada (Central).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

China (Beijing)

The following controls are not supported in China (Beijing).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.13 – Ensure MFA is enabled for the root user ” (p. 355)

the section called "1.14 – Ensure hardware MFA is enabled for the root user " (p. 356)

the section called "PCI DSS" (p. 398)

the section called "[PCI.GuardDuty.1] GuardDuty should be enabled" (p. 423)

the section called "[PCI.IAM.4] Hardware MFA should be enabled for the root user" (p. 427)

the section called "[PCI.IAM.5] Virtual MFA should be enabled for the root user" (p. 428)

the section called "[PCI.Lambda.1] Lambda functions should prohibit public access" (p. 433)

the section called "[PCI.Lambda.2] Lambda functions should be in a VPC" (p. 435)

the section called "[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access" (p. 451)

the section called "AWS Foundational Security Best Practices standard" (p. 457)

the section called "[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period" (p. 460)

the section called "[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication" (p. 462)

the section called "[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled" (p. 462)

the section called "[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL" (p. 463)

the section called "[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones" (p. 465)

the section called "[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)" (p. 465)

the section called "[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1" (p. 466)

the section called "[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones" (p. 468)

the section called "[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)" (p. 469)

the section called "[CloudFront.1] CloudFront distributions should have a default root object configured" (p. 470)

the section called "[CloudFront.2] CloudFront distributions should have origin access identity enabled" (p. 470)

the section called "[CloudFront.3] CloudFront distributions should require encryption in transit" (p. 471)

the section called "[CloudFront.4] CloudFront distributions should have origin failover configured" (p. 471)

the section called "[CloudFront.5] CloudFront distributions should have logging enabled" (p. 471)

the section called "[CloudFront.6] CloudFront distributions should have AWS WAF enabled" (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.4] CodeBuild project environments should have a logging configuration” (p. 481)

the section called “[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled” (p. 481)

the section called “[EC2.15] EC2 subnets should not automatically assign public IP addresses” (p. 494)

the section called “[EC2.16] Unused network access control lists should be removed” (p. 495)

the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)

the section called “[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389” (p. 498)

the section called “[EC2.22] Unused EC2 security groups should be removed” (p. 499)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[EC2.27] Running EC2 Instances should not use key pairs” (p. 501)

the section called “[ECR.1] ECR private repositories should have image scanning configured” (p. 502)

the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)

the section called “[ECR.3] ECR repositories should have at least one lifecycle policy configured” (p. 503)

the section called “[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions” (p. 503)

the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)

the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)

the section called “[ECS.5] ECS containers should be limited to read-only access to root filesystems” (p. 506)

the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)

- the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)
- the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)
- the section called “[EFS.3] EFS access points should enforce a root directory” (p. 510)
- the section called “[EFS.4] EFS access points should enforce a user identity” (p. 511)
- the section called “[EKS.2] EKS clusters should run on a supported Kubernetes version” (p. 512)
- the section called “[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled” (p. 512)
- the section called “[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled” (p. 513)
- the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)
- the section called “[ELB.10] Classic Load Balancers should span multiple Availability Zones” (p. 519)
- the section called “[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 520)
- the section called “[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones” (p. 520)
- the section called “[ELB.14] Classic Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 521)
- the section called “[ES.3] Elasticsearch domains should encrypt data sent between nodes” (p. 524)
- the section called “[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled” (p. 525)
- the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)
- the section called “[IAM.6] Hardware MFA should be enabled for the root user” (p. 533)
- the section called “[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services” (p. 536)
- the section called “[Kinesis.1] Kinesis Data Streams should be encrypted at rest” (p. 537)
- the section called “[Lambda.1] Lambda function policies should prohibit public access” (p. 540)
- the section called “[Lambda.2] Lambda functions should use supported runtimes” (p. 541)
- the section called “[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone” (p. 542)
- the section called “[Network Firewall.3] Network Firewall policies should have at least one rule group associated” (p. 543)
- the section called “[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets” (p. 543)
- the section called “[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets” (p. 544)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[RDS.7] RDS clusters should have deletion protection enabled” (p. 554)

the section called “[RDS.10] IAM authentication should be configured for RDS instances” (p. 558)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.13] RDS automatic minor version upgrades should be enabled” (p. 560)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.16] RDS DB clusters should be configured to copy tags to snapshots” (p. 563)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[RDS.25] RDS database instances should use a custom administrator username” (p. 569)

the section called “[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing” (p. 573)

the section called “[Redshift.8] Amazon Redshift clusters should not use the default Admin username” (p. 573)

the section called “[Redshift.9] Redshift clusters should not use the default database name” (p. 574)

the section called “[S3.8] S3 Block Public Access setting should be enabled at the bucket level” (p. 580)

the section called “[S3.11] S3 buckets should have event notifications enabled” (p. 582)

the section called “[S3.12] S3 access control lists (ACLs) should not be used to manage user access to buckets” (p. 582)

the section called “[S3.13] S3 buckets should have lifecycle policies configured” (p. 583)

the section called “[SageMaker.1] SageMaker notebook instances should not have direct internet access” (p. 584)

the section called “[SecretsManager.3] Remove unused Secrets Manager secrets” (p. 586)

the section called “[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days” (p. 587)

the section called “[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic” (p. 589)

the section called “[SSM.4] SSM documents should not be public” (p. 593)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.2] A WAF Regional rule should have at least one condition” (p. 595)

the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)

the section called “[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group” (p. 596)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

China (Ningxia)

The following controls are not supported in China (Ningxia).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.13 – Ensure MFA is enabled for the root user ” (p. 355)

the section called “1.14 – Ensure hardware MFA is enabled for the root user ” (p. 356)

the section called “PCI DSS” (p. 398)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.IAM.4] Hardware MFA should be enabled for the root user” (p. 427)

the section called “[PCI.IAM.5] Virtual MFA should be enabled for the root user” (p. 428)

the section called “[PCI.Lambda.1] Lambda functions should prohibit public access” (p. 433)

the section called “[PCI.Lambda.2] Lambda functions should be in a VPC” (p. 435)

the section called “[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access” (p. 451)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period” (p. 460)

the section called “[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication” (p. 462)

the section called “[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled” (p. 462)

the section called “[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL” (p. 463)

the section called “[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones” (p. 465)

the section called “[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)” (p. 465)

the section called “[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1” (p. 466)

the section called “[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones” (p. 468)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.4] CodeBuild project environments should have a logging configuration” (p. 481)

the section called “[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled” (p. 481)

the section called “[EC2.15] EC2 subnets should not automatically assign public IP addresses” (p. 494)

the section called “[EC2.16] Unused network access control lists should be removed” (p. 495)

the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)

the section called “[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389” (p. 498)

the section called “[EC2.22] Unused EC2 security groups should be removed” (p. 499)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[EC2.27] Running EC2 Instances should not use key pairs” (p. 501)

the section called “[ECR.1] ECR private repositories should have image scanning configured” (p. 502)

the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)

the section called “[ECR.3] ECR repositories should have at least one lifecycle policy configured” (p. 503)

the section called “[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions” (p. 503)

the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)

the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)

the section called “[ECS.5] ECS containers should be limited to read-only access to root filesystems” (p. 506)

the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)

the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)

the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)

the section called “[EFS.3] EFS access points should enforce a root directory” (p. 510)

the section called “[EFS.4] EFS access points should enforce a user identity” (p. 511)

the section called “[EKS.2] EKS clusters should run on a supported Kubernetes version” (p. 512)

the section called “[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled” (p. 512)

the section called “[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled” (p. 513)

the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)

the section called “[ELB.10] Classic Load Balancers should span multiple Availability Zones” (p. 519)

the section called “[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 520)

the section called “[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones” (p. 520)

the section called “[ELB.14] Classic Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 521)

the section called “[ES.3] Elasticsearch domains should encrypt data sent between nodes” (p. 524)

the section called “[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled” (p. 525)

the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)

the section called “[IAM.6] Hardware MFA should be enabled for the root user” (p. 533)

the section called “[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services” (p. 536)

the section called “[Kinesis.1] Kinesis Data Streams should be encrypted at rest” (p. 537)

the section called “[Lambda.1] Lambda function policies should prohibit public access” (p. 540)

the section called “[Lambda.2] Lambda functions should use supported runtimes” (p. 541)

the section called “[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone” (p. 542)

the section called “[Network Firewall.3] Network Firewall policies should have at least one rule group associated” (p. 543)

the section called “[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets” (p. 543)

the section called “[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets” (p. 544)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[RDS.7] RDS clusters should have deletion protection enabled” (p. 554)

the section called “[RDS.9] Database logging should be enabled” (p. 555)

the section called “[RDS.10] IAM authentication should be configured for RDS instances” (p. 558)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.13] RDS automatic minor version upgrades should be enabled” (p. 560)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[RDS.25] RDS database instances should use a custom administrator username” (p. 569)

the section called “[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled” (p. 571)

the section called “[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing” (p. 573)

the section called “[Redshift.8] Amazon Redshift clusters should not use the default Admin username” (p. 573)

the section called “[Redshift.9] Redshift clusters should not use the default database name” (p. 574)

the section called “[S3.8] S3 Block Public Access setting should be enabled at the bucket level” (p. 580)

the section called “[S3.11] S3 buckets should have event notifications enabled” (p. 582)

the section called “[S3.12] S3 access control lists (ACLs) should not be used to manage user access to buckets” (p. 582)

the section called “[S3.13] S3 buckets should have lifecycle policies configured” (p. 583)

the section called “[SageMaker.1] SageMaker notebook instances should not have direct internet access” (p. 584)

the section called “[SecretsManager.3] Remove unused Secrets Manager secrets” (p. 586)

the section called “[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days” (p. 587)

the section called “[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic” (p. 589)

the section called “[SSM.4] SSM documents should not be public” (p. 593)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.2] A WAF Regional rule should have at least one condition” (p. 595)

the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)

the section called “[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group” (p. 596)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Europe (Frankfurt)

The following controls are not supported in Europe (Frankfurt).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Europe (Ireland)

The following controls are not supported in Europe (Ireland).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Europe (London)

The following controls are not supported in Europe (London).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Europe (Milan)

The following controls are not supported in Europe (Milan).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.4 – Ensure access keys are rotated every 90 days or less ” (p. 351)

the section called “1.20 - Ensure a support role has been created to manage incidents with AWS Support ” (p. 358)

the section called “4.1 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 22 ” (p. 395)

the section called “4.2 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389 ” (p. 396)

the section called “PCI DSS” (p. 398)

the section called “[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 405)

the section called “[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 406)

the section called “[PCI.DMS.1] AWS Database Migration Service replication instances should not be public” (p. 411)

the section called “[PCI.EC2.1] Amazon EBS snapshots should not be publicly restorable” (p. 413)

the section called “[PCI.EC2.4] Unused EC2 EIPs should be removed” (p. 416)

the section called “[PCI.EC2.5] Security groups should not allow ingress from 0.0.0.0/0 to port 22” (p. 417)

the section called “[PCI.ELBV2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 419)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.RDS.1] Amazon RDS snapshots should prohibit public access” (p. 439)

the section called “[PCI.S3.6] S3 Block Public Access setting should be enabled” (p. 449)

the section called “[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access” (p. 451)

the section called “[PCI.SSM.1] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation” (p. 452)

the section called “[PCI.SSM.2] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 454)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period” (p. 460)

the section called “[APIGateway.1] API Gateway REST and WebSocket API logging should be enabled” (p. 461)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 479)

the section called “[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 480)

the section called “[DMS.1] AWS Database Migration Service replication instances should not be public” (p. 483)

the section called “[EC2.1] Amazon EBS snapshots should not be public, determined by the availability to be restorable by anyone” (p. 485)

the section called “[EC2.3] Attached EBS volumes should be encrypted at rest” (p. 487)

the section called “[EC2.4] Stopped EC2 instances should be removed after a specified time period” (p. 488)

the section called “[EC2.8] EC2 instances should use IMDSv2” (p. 490)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[EFS.1] Amazon EFS should be configured to encrypt file data at rest using AWS KMS” (p. 509)

the section called “[EFS.2] Amazon EFS volumes should be in backup plans” (p. 510)

the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)

the section called “[ELB.4] Application load balancers should be configured to drop HTTP headers” (p. 515)

the section called “[ELB.8] Classic Load Balancers with HTTPS/SSL listeners should use a predefined security policy that has strong configuration” (p. 518)

the section called “[ELBv2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS” (p. 522)

the section called “[EMR.1] Amazon EMR cluster master nodes should not have public IP addresses” (p. 523)

the section called “[ES.3] Elasticsearch domains should encrypt data sent between nodes” (p. 524)

the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)

the section called “[IAM.3] IAM users' access keys should be rotated every 90 days or less” (p. 530)

the section called “[KMS.3] AWS KMS keys should not be unintentionally deleted” (p. 539)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[RDS.1] RDS snapshots should be private” (p. 550)

the section called “[RDS.9] Database logging should be enabled” (p. 555)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit” (p. 570)

the section called “[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled” (p. 571)

the section called “[S3.1] S3 Block Public Access setting should be enabled” (p. 574)

the section called “[SageMaker.1] SageMaker notebook instances should not have direct internet access” (p. 584)

the section called “[SSM.2] All EC2 instances managed by Systems Manager should be compliant with patching requirements” (p. 591)

the section called “[SSM.3] Instances managed by Systems Manager should have an association compliance status of COMPLIANT” (p. 592)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Europe (Paris)

The following controls are not supported in Europe (Paris).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Europe (Stockholm)

The following controls are not supported in Europe (Stockholm).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Middle East (Bahrain)

The following controls are not supported in Middle East (Bahrain).

the section called “PCI DSS” (p. 398)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.S3.6] S3 Block Public Access setting should be enabled” (p. 449)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)

the section called “[RDS.7] RDS clusters should have deletion protection enabled” (p. 554)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.16] RDS DB clusters should be configured to copy tags to snapshots” (p. 563)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled” (p. 572)

the section called “[S3.1] S3 Block Public Access setting should be enabled” (p. 574)

the section called “[SSM.2] All EC2 instances managed by Systems Manager should be compliant with patching requirements” (p. 591)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

South America (São Paulo)

The following controls are not supported in South America (São Paulo).

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[RDS.7] RDS clusters should have deletion protection enabled” (p. 554)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.16] RDS DB clusters should be configured to copy tags to snapshots” (p. 563)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

AWS GovCloud (US-East)

The following controls are not supported in AWS GovCloud (US-East).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.13 – Ensure MFA is enabled for the root user ” (p. 355)

the section called “1.14 – Ensure hardware MFA is enabled for the root user ” (p. 356)

the section called “PCI DSS” (p. 398)

the section called “[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 405)

the section called “[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 406)

the section called “[PCI.GuardDuty.1] GuardDuty should be enabled” (p. 423)

the section called “[PCI.IAM.4] Hardware MFA should be enabled for the root user” (p. 427)

the section called “[PCI.IAM.5] Virtual MFA should be enabled for the root user” (p. 428)

the section called “[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access” (p. 451)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication” (p. 462)

the section called “[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled” (p. 462)

the section called “[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL” (p. 463)

the section called “[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones” (p. 465)

the section called “[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)” (p. 465)

the section called “[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1” (p. 466)

the section called “[AutoScaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses” (p. 467)

the section called “[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones” (p. 468)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 479)

the section called “[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 480)

the section called “[CodeBuild.4] CodeBuild project environments should have a logging configuration” (p. 481)

the section called “[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled” (p. 481)

the section called “[DynamoDB.1] DynamoDB tables should automatically scale capacity with demand” (p. 484)

the section called “[EC2.15] EC2 subnets should not automatically assign public IP addresses” (p. 494)

the section called “[EC2.16] Unused network access control lists should be removed” (p. 495)

the section called “[EC2.17] EC2 instances should not use multiple ENIs” (p. 495)

the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)

the section called “[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389” (p. 498)

the section called “[EC2.22] Unused EC2 security groups should be removed” (p. 499)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[EC2.27] Running EC2 Instances should not use key pairs” (p. 501)

the section called “[ECR.1] ECR private repositories should have image scanning configured” (p. 502)

the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)

the section called “[ECR.3] ECR repositories should have at least one lifecycle policy configured” (p. 503)

the section called “[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions” (p. 503)

the section called “[ECS.2] Amazon ECS services should not have public IP addresses assigned to them automatically” (p. 504)

the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)

the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)

the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)

the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)

the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)

the section called “[EFS.2] Amazon EFS volumes should be in backup plans” (p. 510)

the section called “[EFS.3] EFS access points should enforce a root directory” (p. 510)

the section called “[EFS.4] EFS access points should enforce a user identity” (p. 511)

the section called “[EKS.2] EKS clusters should run on a supported Kubernetes version” (p. 512)

the section called “[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled” (p. 512)

the section called “[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled” (p. 513)

the section called “[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager” (p. 514)

the section called “[ELB.8] Classic Load Balancers with HTTPS/SSL listeners should use a predefined security policy that has strong configuration” (p. 518)

the section called “[ELB.10] Classic Load Balancers should span multiple Availability Zones” (p. 519)

the section called “[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 520)

the section called “[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones” (p. 520)

the section called “[ELB.14] Classic Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 521)

the section called “[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled” (p. 525)

the section called “[GuardDuty.1] GuardDuty should be enabled” (p. 528)

the section called “[IAM.6] Hardware MFA should be enabled for the root user” (p. 533)

the section called “[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services” (p. 536)

the section called “[Kinesis.1] Kinesis Data Streams should be encrypted at rest” (p. 537)

the section called “[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone” (p. 542)

the section called “[Network Firewall.3] Network Firewall policies should have at least one rule group associated” (p. 543)

the section called “[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets” (p. 543)

the section called “[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets” (p. 544)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.13] RDS automatic minor version upgrades should be enabled” (p. 560)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[RDS.25] RDS database instances should use a custom administrator username” (p. 569)

the section called “[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing” (p. 573)

the section called “[Redshift.8] Amazon Redshift clusters should not use the default Admin username” (p. 573)

the section called “[Redshift.9] Redshift clusters should not use the default database name” (p. 574)

the section called “[S3.8] S3 Block Public Access setting should be enabled at the bucket level” (p. 580)

the section called “[S3.11] S3 buckets should have event notifications enabled” (p. 582)

the section called “[S3.12] S3 access control lists (ACLs) should not be used to manage user access to buckets” (p. 582)

the section called “[S3.13] S3 buckets should have lifecycle policies configured” (p. 583)

the section called “[SecretsManager.3] Remove unused Secrets Manager secrets” (p. 586)

the section called “[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days” (p. 587)

the section called “[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic” (p. 589)

the section called “[SSM.4] SSM documents should not be public” (p. 593)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.2] A WAF Regional rule should have at least one condition” (p. 595)

the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)

the section called “[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group” (p. 596)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

AWS GovCloud (US-West)

The following controls are not supported in AWS GovCloud (US-West).

the section called “CIS AWS Foundations Benchmark” (p. 346)

the section called “1.13 – Ensure MFA is enabled for the root user ” (p. 355)

the section called “1.14 – Ensure hardware MFA is enabled for the root user ” (p. 356)

the section called “PCI DSS” (p. 398)

the section called “[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 405)

the section called “[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 406)

the section called “[PCI.IAM.4] Hardware MFA should be enabled for the root user” (p. 427)

the section called “[PCI.IAM.5] Virtual MFA should be enabled for the root user” (p. 428)

the section called “AWS Foundational Security Best Practices standard” (p. 457)

the section called “[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication” (p. 462)

the section called “[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled” (p. 462)

the section called “[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL” (p. 463)

the section called “[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones” (p. 465)

the section called “[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)” (p. 465)

the section called “[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1” (p. 466)

the section called “[AutoScaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses” (p. 467)

the section called “[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones” (p. 468)

the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469)

the section called “[CloudFront.1] CloudFront distributions should have a default root object configured” (p. 470)

the section called “[CloudFront.2] CloudFront distributions should have origin access identity enabled” (p. 470)

the section called “[CloudFront.3] CloudFront distributions should require encryption in transit” (p. 471)

the section called “[CloudFront.4] CloudFront distributions should have origin failover configured” (p. 471)

the section called “[CloudFront.5] CloudFront distributions should have logging enabled” (p. 471)

the section called “[CloudFront.6] CloudFront distributions should have AWS WAF enabled” (p. 472)

the section called “[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates” (p. 473)

the section called “[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests” (p. 473)

the section called “[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins” (p. 474)

the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474)

the section called “[CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth” (p. 479)

the section called “[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials” (p. 480)

the section called “[CodeBuild.4] CodeBuild project environments should have a logging configuration” (p. 481)

the section called “[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled” (p. 481)

the section called “[DynamoDB.1] DynamoDB tables should automatically scale capacity with demand” (p. 484)

the section called “[EC2.15] EC2 subnets should not automatically assign public IP addresses” (p. 494)

the section called “[EC2.16] Unused network access control lists should be removed” (p. 495)

the section called “[EC2.17] EC2 instances should not use multiple ENIs” (p. 495)

the section called “[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up” (p. 498)

the section called “[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389” (p. 498)

the section called “[EC2.22] Unused EC2 security groups should be removed” (p. 499)

the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499)

the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500)

the section called “[EC2.27] Running EC2 Instances should not use key pairs” (p. 501)

the section called “[ECR.1] ECR private repositories should have image scanning configured” (p. 502)

the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)

the section called “[ECR.3] ECR repositories should have at least one lifecycle policy configured” (p. 503)

the section called “[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions” (p. 503)

the section called “[ECS.2] Amazon ECS services should not have public IP addresses assigned to them automatically” (p. 504)

the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)

the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)

the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)

the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)

the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)

the section called “[EFS.2] Amazon EFS volumes should be in backup plans” (p. 510)

the section called “[EFS.3] EFS access points should enforce a root directory” (p. 510)

the section called “[EFS.4] EFS access points should enforce a user identity” (p. 511)

the section called “[EKS.2] EKS clusters should run on a supported Kubernetes version” (p. 512)

the section called “[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled” (p. 512)

the section called “[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled” (p. 513)

the section called “[ELB.10] Classic Load Balancers should span multiple Availability Zones” (p. 519)

the section called “[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 520)

the section called “[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones” (p. 520)

the section called “[ELB.14] Classic Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 521)

the section called “[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled” (p. 525)

the section called “[IAM.6] Hardware MFA should be enabled for the root user” (p. 533)

the section called “[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services” (p. 536)

the section called “[Kinesis.1] Kinesis Data Streams should be encrypted at rest” (p. 537)

the section called “[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone” (p. 542)

the section called “[Network Firewall.3] Network Firewall policies should have at least one rule group associated” (p. 543)

the section called “[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets” (p. 543)

the section called “[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets” (p. 544)

the section called “[OpenSearch.1] OpenSearch domains should have encryption at rest enabled” (p. 545)

the section called “[OpenSearch.2] OpenSearch domains should be in a VPC” (p. 546)

the section called “[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes” (p. 546)

the section called “[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled” (p. 547)

the section called “[OpenSearch.5] OpenSearch domains should have audit logging enabled” (p. 547)

the section called “[OpenSearch.6] OpenSearch domains should have at least three data nodes” (p. 548)

the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)

the section called “[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2” (p. 549)

the section called “[RDS.12] IAM authentication should be configured for RDS clusters” (p. 559)

the section called “[RDS.13] RDS automatic minor version upgrades should be enabled” (p. 560)

the section called “[RDS.14] Amazon Aurora clusters should have backtracking enabled” (p. 561)

the section called “[RDS.15] RDS DB clusters should be configured for multiple Availability Zones” (p. 562)

the section called “[RDS.24] RDS database clusters should use a custom administrator username” (p. 569)

the section called “[RDS.25] RDS database instances should use a custom administrator username” (p. 569)

the section called “[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing” (p. 573)

the section called “[Redshift.8] Amazon Redshift clusters should not use the default Admin username” (p. 573)

the section called “[Redshift.9] Redshift clusters should not use the default database name” (p. 574)

the section called “[S3.8] S3 Block Public Access setting should be enabled at the bucket level” (p. 580)

the section called “[S3.11] S3 buckets should have event notifications enabled” (p. 582)

the section called “[S3.12] S3 access control lists (ACLs) should not be used to manage user access to buckets” (p. 582)

the section called “[S3.13] S3 buckets should have lifecycle policies configured” (p. 583)

the section called “[SecretsManager.3] Remove unused Secrets Manager secrets” (p. 586)

the section called “[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days” (p. 587)

the section called “[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic” (p. 589)

the section called “[SSM.4] SSM documents should not be public” (p. 593)

the section called “[WAF.1] AWS WAF Classic global web ACL logging should be enabled” (p. 594)

the section called “[WAF.2] A WAF Regional rule should have at least one condition” (p. 595)

the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)

the section called “[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group” (p. 596)

the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597)

the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597)

the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598)

Setting up AWS Security Hub

Whether an account needs to enable AWS Security Hub manually depends on how the accounts are managed.

You can use the integration with AWS Organizations, or you can manage accounts manually.

In both cases, you set up Security Hub and manage accounts separately in each Region. All accounts also must enable AWS Config, which is needed for the security checks against security controls. See [the section called "Enabling AWS Config" \(p. 8\)](#).

Organizations integration

If you use the integration with AWS Organizations, then most organization accounts have Security Hub enabled automatically.

The organization management account chooses a Security Hub administrator account. Security Hub is enabled automatically for the chosen account. See [the section called "Designating a Security Hub administrator account" \(p. 93\)](#).

The Security Hub administrator account enables organization accounts as member accounts. Those organization accounts also have Security Hub enabled automatically. See [the section called "Managing organization member accounts" \(p. 98\)](#).

The only organization account for which Security Hub is not enabled automatically is the organization management account. The organization management account does not need to enable Security Hub before it designates the Security Hub administrator account. The organization management account must enable Security Hub before it is enabled as a member account.

Manual account management

Accounts that are not managed using the Organizations integration must enable Security Hub manually.

The Security Hub administrator-member relationship is established when the member account accepts an invitation from the administrator account. See [the section called "Managing member accounts by invitation" \(p. 102\)](#).

Contents

- [Enabling Security Hub manually \(p. 66\)](#)
 - [Attaching the required IAM policy to the IAM identity \(p. 67\)](#)
 - [Enabling Security Hub \(console\) \(p. 67\)](#)
 - [Enabling Security Hub \(Security Hub API, AWS CLI\) \(p. 67\)](#)
 - [Enabling Security Hub \(Multi-account script\) \(p. 68\)](#)
- [Service-linked role assigned to Security Hub \(p. 68\)](#)

Enabling Security Hub manually

After you attach the required policy to the IAM identity, you use that identity to enable Security Hub. You can enable Security Hub from the AWS Management Console or the API.

Security Hub also provides a script in GitHub that allows you to enable multiple accounts across Regions.

Attaching the required IAM policy to the IAM identity

The IAM identity (user, role, or group) that you use to enable Security Hub must have the required permissions.

If you enable the integration with AWS Organizations, then accounts in your organization have Security Hub enabled automatically. The required permissions also are handled automatically.

Accounts that are not managed using Organizations must enable Security Hub manually. The IAM identity (user, role, or group) that you use to enable Security Hub must have the required permissions.

To grant the permissions required to enable Security Hub, attach the Security Hub managed policy [AWSSecurityHubFullAccess](#) (p. 82) to an IAM user, group, or role.

Enabling Security Hub (console)

When you enable Security Hub from the console, you also have the option to enable the supported security standards.

To enable Security Hub

1. Use the credentials of the IAM identity to sign in to the Security Hub console.
2. When you open the Security Hub console for the first time, choose **Enable AWS Security Hub**.
3. On the welcome page, **Security standards** lists the security standards that Security Hub supports.

To enable a standard, select its check box.

To disable a standard, clear its check box.

You can enable or disable a standard or its individual controls at any time. For information about the security standards and how to manage them, see [Standards and controls](#) (p. 323).

4. Choose **Enable Security Hub**.

Enabling Security Hub (Security Hub API, AWS CLI)

To enable Security Hub, you can use an API call or the AWS Command Line Interface.

To enable Security Hub (Security HubAPI, AWS CLI)

- **Security Hub API** – Use the [EnableSecurityHub](#) operation. When you enable Security Hub from the API, it automatically enables these security standards.
 - CIS AWS Foundations Benchmark
 - AWS Foundational Security Best Practices Standard

If you do not want to enable these standards, then set `EnableDefaultStandards` to `false`.

You can also use the `Tags` parameter to assign tag values to the hub resource.

- **AWS CLI** – At the command line, run the [enable-security-hub](#) command. To enable the default standards, include `--enable-default-standards`. To not enable the default standards, include `--no-enable-default-standards`.

```
aws securityhub enable-security-hub [--tags <tag values>] [--enable-default-standards | --no-enable-default-standards]
```

Example

```
aws securityhub enable-security-hub --enable-default-standards --tags '{"Department":  
"Security"}'
```

After you enable Security Hub, you can enable or disable standards. See [the section called “Disabling or enabling a security standard” \(p. 333\)](#).

Enabling Security Hub (Multi-account script)

The [Security Hub multi-account enablement script in GitHub](#) allows you to enable Security Hub across accounts and Regions. The script also automates the process of sending invitations to member accounts and enabling AWS Config.

The script automatically enables resource recording for all resources, including global resources, in all Regions. It does not limit recording of global resources to a single Region.

There is a corresponding script to disable Security Hub across accounts and Regions.

The readme file provides details on how to use the script. It includes the following information:

- How to add the required IAM policy to the accounts
- How to configure the execution environment
- How to run the script

Service-linked role assigned to Security Hub

When you enable Security Hub, it is assigned a service-linked role named `AWSServiceRoleForSecurityHub`. This service-linked role includes the permissions and trust policy that Security Hub requires to do the following:

- Detect and aggregate findings from Amazon GuardDuty, Amazon Inspector, and Amazon Macie
- Configure the requisite AWS Config infrastructure to run security checks for the supported standards (in this release, CIS AWS Foundations)

To view the details of `AWSServiceRoleForSecurityHub`, on the **Settings** page, choose **General** and then **View service permissions**. For more information, see [Using service-linked roles for AWS Security Hub \(p. 80\)](#).

For more information about service-linked roles, see [Using service-linked roles](#) in the *IAM User Guide*.

Security in AWS Security Hub

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The [shared responsibility model](#) describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the [AWS compliance programs](#). To learn about the compliance programs that apply to AWS Security Hub, see [AWS Services in Scope by Compliance Program](#).
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Security Hub. The following topics show you how to configure Security Hub to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Security Hub resources.

Topics

- [Data protection in AWS Security Hub \(p. 69\)](#)
- [AWS Identity and Access Management for AWS Security Hub \(p. 70\)](#)
- [Using service-linked roles for AWS Security Hub \(p. 80\)](#)
- [AWS managed policies for AWS Security Hub \(p. 81\)](#)
- [Compliance validation for AWS Security Hub \(p. 86\)](#)
- [Infrastructure security in AWS Security Hub \(p. 86\)](#)
- [AWS Security Hub and interface VPC endpoints \(AWS PrivateLink\) \(p. 87\)](#)

Data protection in AWS Security Hub

The AWS [shared responsibility model](#) applies to data protection in AWS Security Hub. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the [Data Privacy FAQ](#). For information about data protection in Europe, see the [AWS Shared Responsibility Model and GDPR](#) blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM). That way each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.

- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with Security Hub or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Security Hub is a multi-tenant service offering. To ensure data protection, Security Hub encrypts data at rest and data in transit between component services.

AWS Identity and Access Management for AWS Security Hub

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Security Hub resources. IAM is an AWS service that you can use with no additional charge.

Topics

- [Audience \(p. 70\)](#)
- [Authenticating with identities \(p. 71\)](#)
- [AWS account root user \(p. 71\)](#)
- [IAM users and groups \(p. 71\)](#)
- [IAM roles \(p. 72\)](#)
- [Managing access using policies \(p. 72\)](#)
- [How AWS Security Hub works with IAM \(p. 74\)](#)

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Security Hub.

Service user – If you use the Security Hub service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Security Hub features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Security Hub, see [Troubleshooting AWS Security Hub identity and access \(p. 78\)](#).

Service administrator – If you're in charge of Security Hub resources at your company, you probably have full access to Security Hub. It's your job to determine which Security Hub features and resources your employees should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Security Hub, see [How AWS Security Hub works with IAM \(p. 74\)](#).

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Security Hub. To view example Security Hub identity-based policies that you can use in IAM, see [AWS Security Hub identity-based policy examples \(p. 77\)](#).

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. For more information about signing in using the AWS Management Console, see [Signing in to the AWS Management Console as an IAM user or root user](#) in the *IAM User Guide*.

You must be *authenticated* (signed in to AWS) as the AWS account root user, an IAM user, or by assuming an IAM role. You can also use your company's single sign-on authentication or even sign in using Google or Facebook. In these cases, your administrator previously set up identity federation using IAM roles. When you access AWS using credentials from another company, you are assuming a role indirectly.

To sign in directly to the [AWS Management Console](#), use your password with your root user email address or your IAM user name. You can access AWS programmatically using your root user or IAM users access keys. AWS provides SDK and command line tools to cryptographically sign your request using your credentials. If you don't use AWS tools, you must sign the request yourself. Do this using *Signature Version 4*, a protocol for authenticating inbound API requests. For more information about authenticating requests, see [Signature Version 4 signing process](#) in the *AWS General Reference*.

Regardless of the authentication method that you use, you might also be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.

AWS account root user

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the [best practice of using the root user only to create your first IAM user](#). Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. An IAM user can have long-term credentials such as a user name and password or a set of access keys. To learn how to generate access keys, see [Managing access keys for IAM users](#) in the *IAM User Guide*. When you generate access keys for an IAM user, make sure you view and securely save the key pair. You cannot recover the secret access key in the future. Instead, you must generate a new access key pair.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see [When to create an IAM user \(instead of a role\)](#) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by [switching roles](#). You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see [Using IAM roles](#) in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Temporary IAM user permissions** – An IAM user can assume an IAM role to temporarily take on different permissions for a specific task.
- **Federated user access** – Instead of creating an IAM user, you can use existing identities from AWS Directory Service, your enterprise user directory, or a web identity provider. These are known as *federated users*. AWS assigns a role to a federated user when access is requested through an [identity provider](#). For more information about federated users, see [Federated users and roles](#) in the *IAM User Guide*.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
- **Principal permissions** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. Policies grant permissions to a principal. When you use some services, you might perform an action that then triggers another action in a different service. In this case, you must have permissions to perform both actions. To see whether an action requires additional dependent actions in a policy, see [Actions, resources, and condition keys for AWS Security Hub](#) in the *Service Authorization Reference*.
- **Service role** – A service role is an [IAM role](#) that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see [Creating a role to delegate permissions to an AWS service](#) in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see [Using an IAM role to grant permissions to applications running on Amazon EC2 instances](#) in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see [When to create an IAM role \(instead of a user\)](#) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to IAM identities or AWS resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions.

You can sign in as the root user or an IAM user, or you can assume an IAM role. When you then make a request, AWS evaluates the related identity-based or resource-based policies. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see [Overview of JSON policies](#) in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

Every IAM entity (user or role) starts with no permissions. In other words, by default, users can do nothing, not even change their own password. To give a user permission to do something, an administrator must attach a permissions policy to a user. Or the administrator can add the user to a group that has the intended permissions. When an administrator gives permissions to a group, all users in that group are granted those permissions.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see [Creating IAM policies](#) in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see [Choosing between managed policies and inline policies](#) in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must [specify a principal](#) in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see [Permissions boundaries for IAM entities](#) in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for

grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see [How SCPs work](#) in the *AWS Organizations User Guide*.

- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see [Session policies](#) in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see [Policy evaluation logic](#) in the *IAM User Guide*.

How AWS Security Hub works with IAM

Before you use IAM to manage access to Security Hub, you should understand what IAM features are available to use with Security Hub. To get a high-level view of how Security Hub and other AWS services work with IAM, see [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Topics

- [Security Hub identity-based policies \(p. 74\)](#)
- [Security Hub resource-based policies \(Not supported\) \(p. 76\)](#)
- [Authorization based on Security Hub tags \(p. 76\)](#)
- [Security Hub IAM roles \(p. 76\)](#)
- [Service-linked roles \(p. 76\)](#)
- [Service roles \(p. 76\)](#)
- [AWS Security Hub identity-based policy examples \(p. 77\)](#)

Security Hub identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. Security Hub supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see [IAM JSON Policy Elements Reference](#) in the *IAM User Guide*.

Actions

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

Policy actions in Security Hub use the following prefix before the action: `securityhub:`. For example, to grant a user permission to enable Security Hub using the `EnableSecurityHub` API operation, you

include the `securityhub:EnableSecurityHub` action in the policy assigned to that user. Policy statements must include either an `Action` or `NotAction` element. Security Hub defines its own set of actions that describe tasks that you can perform with this service.

To specify multiple actions in a single statement, separate them with commas as follows:

```
"Action": [
  "securityhub:action1",
  "securityhub:action2"
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `Get`, include the following line in your policy:

```
"Action": "securityhub:Get*"
```

To see a list of Security Hub actions, see [Actions defined by AWS Security Hub](#) in the *Service Authorization Reference*.

Resources

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its [Amazon Resource Name \(ARN\)](#). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*" 
```

For more information about the format of ARNs, see [Amazon Resource Names \(ARNs\) and AWS Service Namespaces](#).

To see a list of Security Hub resource types and their ARNs, see [Resource types defined by AWS Security Hub](#) in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see [Actions defined by AWS Security Hub](#).

Condition keys

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or *Condition block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use [condition operators](#), such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical AND operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical OR operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see [IAM policy elements: variables and tags](#) in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see [AWS global condition context keys](#) in the *IAM User Guide*.

Security Hub defines its own set of condition keys and also supports using some global condition keys. To see all AWS global condition keys, see [AWS Global Condition Context Keys](#) in the *IAM User Guide*.

Security Hub actions support the `securityhub:TargetAccount` condition key.

To control access to [BatchUpdateFindings](#), Security Hub supports the `securityhub:ASFFSyntaxPath` condition key. For details on configuring access to [BatchUpdateFindings](#), see the section called “Configuring access to [BatchUpdateFindings](#)” (p. 123).

To see a list of Security Hub condition keys, see [Condition keys for AWS Security Hub](#) in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see [Actions defined by AWS Security Hub](#).

Security Hub resource-based policies (Not supported)

Security Hub does not support resource-based policies.

Authorization based on Security Hub tags

You can add tags to Security Hub resources or pass tags in a request to Security Hub. To control access based on tags, you provide tag information in the [condition element](#) of a policy using the `securityhub:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

Security Hub IAM roles

An [IAM role](#) is an entity within your AWS account that has specific permissions.

Using temporary credentials with Security Hub

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS STS API operations such as [AssumeRole](#) or [GetFederationToken](#).

Security Hub supports using temporary credentials.

Service-linked roles

[Service-linked roles](#) allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Security Hub supports service-linked roles.

Service roles

This feature allows a service to assume a [service role](#) on your behalf. This role allows the service to access resources in other services to complete an action on your behalf. Service roles appear in your IAM account and are owned by the account. This means that an IAM administrator can change the permissions for this role. However, doing so might break the functionality of the service.

Security Hub supports service roles.

AWS Security Hub identity-based policy examples

By default, IAM users and roles don't have permission to create or modify Security Hub resources. They also can't perform tasks using the AWS Management Console, AWS CLI, or AWS API. An IAM administrator must create IAM policies that grant users and roles permission to perform specific API operations on the specified resources they need. The administrator must then attach those policies to the IAM users or groups that require those permissions.

To learn how to create an IAM identity-based policy using these example JSON policy documents, see [Creating Policies on the JSON Tab](#) in the *IAM User Guide*.

Topics

- [Policy best practices \(p. 77\)](#)
- [Using the Security Hub console \(p. 77\)](#)
- [Troubleshooting AWS Security Hub identity and access \(p. 78\)](#)

Policy best practices

Identity-based policies are very powerful. They determine whether someone can create, access, or delete Security Hub resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started using AWS managed policies** – To start using Security Hub quickly, use AWS managed policies to give your employees the permissions they need. These policies are already available in your account and are maintained and updated by AWS. For more information, see [Get started using permissions with AWS managed policies](#) in the *IAM User Guide*.
- **Grant least privilege** – When you create custom policies, grant only the permissions required to perform a task. Start with a minimum set of permissions and grant additional permissions as necessary. Doing so is more secure than starting with permissions that are too lenient and then trying to tighten them later. For more information, see [Grant least privilege](#) in the *IAM User Guide*.
- **Enable MFA for sensitive operations** – For extra security, require IAM users to use multi-factor authentication (MFA) to access sensitive resources or API operations. For more information, see [Using multi-factor authentication \(MFA\) in AWS](#) in the *IAM User Guide*.
- **Use policy conditions for extra security** – To the extent that it's practical, define the conditions under which your identity-based policies allow access to a resource. For example, you can write conditions to specify a range of allowable IP addresses that a request must come from. You can also write conditions to allow requests only within a specified date or time range, or to require the use of SSL or MFA. For more information, see [IAM JSON policy elements: Condition](#) in the *IAM User Guide*.

Using the Security Hub console

To access the AWS Security Hub console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Security Hub resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (IAM users or roles) with that policy.

To ensure that those entities can still use the Security Hub console, also attach the following AWS managed policy to the entities. For more information, see [Adding permissions to a user](#) in the *IAM User Guide*:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
    "Effect": "Allow",
    "Action": "securityhub:*",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Action": "iam:CreateServiceLinkedRole",
    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iam:AWSServiceName": "securityhub.amazonaws.com"
      }
    }
  }
]
}
```

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that you're trying to perform.

Troubleshooting AWS Security Hub identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Security Hub and IAM.

Topics

- [I am not authorized to perform an action in Security Hub \(p. 78\)](#)
- [I am not authorized to perform iam:PassRole \(p. 78\)](#)
- [I want to view my access keys \(p. 79\)](#)
- [I'm an administrator and want to allow others to access Security Hub \(p. 79\)](#)
- [I want to allow people outside My AWS account to access my Security Hub resources \(p. 79\)](#)

I am not authorized to perform an action in Security Hub

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a `widget` but does not have `securityhub:GetWidget` permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
securityhub:GetWidget on resource: my-example-widget
```

In this case, Mateo asks his administrator to update his policies to allow him to access the `my-example-widget` resource using the `securityhub:GetWidget` action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your user name and password. Ask that person to update your policies to allow you to pass a role to Security Hub.

Some AWS services allow you to pass an existing role to that service, instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Security Hub. However, the action requires the service to have permissions granted by a service role. Mary does not have permissions to pass the role to the service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

In this case, Mary asks her administrator to update her policies to allow her to perform the `iam:PassRole` action.

I want to view my access keys

After you create your IAM user access keys, you can view your access key ID at any time. However, you can't view your secret access key again. If you lose your secret key, you must create a new access key pair.

Access keys consist of two parts: an access key ID (for example, `AKIAIOSFODNN7EXAMPLE`) and a secret access key (for example, `wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY`). Like a user name and password, you must use both the access key ID and secret access key together to authenticate your requests. Manage your access keys as securely as you do your user name and password.

Important

Do not provide your access keys to a third party, even to help [find your canonical user ID](#). By doing this, you might give someone permanent access to your account.

When you create an access key pair, you are prompted to save the access key ID and secret access key in a secure location. The secret access key is available only at the time you create it. If you lose your secret access key, you must add new access keys to your IAM user. You can have a maximum of two access keys. If you already have two, you must delete one key pair before creating a new one. To view instructions, see [Managing access keys](#) in the *IAM User Guide*.

I'm an administrator and want to allow others to access Security Hub

To allow others to access Security Hub, you must create an IAM entity (user or role) for the person or application that needs access. They will use the credentials for that entity to access AWS. You must then attach a policy to the entity that grants them the correct permissions in Security Hub.

To get started right away, see [Creating your first IAM delegated user and group](#) in the *IAM User Guide*.

I want to allow people outside My AWS account to access my Security Hub resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Security Hub supports these features, see [How AWS Security Hub works with IAM](#) (p. 74).
- To learn how to provide access to your resources across AWS accounts that you own, see [Providing access to an IAM user in another AWS account that you own](#) in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see [Providing access to AWS accounts owned by third parties](#) in the *IAM User Guide*.
- To learn how to provide access through identity federation, see [Providing access to externally authenticated users \(identity federation\)](#) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see [How IAM roles differ from resource-based policies](#) in the *IAM User Guide*.

Using service-linked roles for AWS Security Hub

AWS Security Hub uses AWS Identity and Access Management (IAM) [service-linked roles](#). A service-linked role is a unique type of IAM role that is linked directly to Security Hub. Service-linked roles are predefined by Security Hub and include all the permissions that Security Hub requires to call other AWS services on your behalf.

A service-linked role makes setting up Security Hub easier because you don't have to manually add the necessary permissions. Security Hub defines the permissions of its service-linked role, and unless the permissions are defined otherwise, only Security Hub can assume the role. The defined permissions include the trust policy and the permissions policy, and you can't attach that permissions policy to any other IAM entity.

Security Hub supports using service-linked roles in all of the Regions where Security Hub is available. For more information, see [Regional limits \(p. 13\)](#).

You can delete the Security Hub service-linked role only after first disabling Security Hub in all Regions where it's enabled. This protects your Security Hub resources because you can't inadvertently remove permissions to access them.

For information about other services that support service-linked roles, see [AWS services that work with IAM](#) in the *IAM User Guide* and locate the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for Security Hub

Security Hub uses the service-linked role named `AWSServiceRoleForSecurityHub`. It's a service-linked role required for AWS Security Hub to access your resources.

The `AWSServiceRoleForSecurityHub` service-linked role trusts the following services to assume the role:

- `securityhub.amazonaws.com`

The `AWSServiceRoleForSecurityHub` service-linked role uses the managed policy [AWSSecurityHubServiceRolePolicy \(p. 84\)](#).

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For the `AWSServiceRoleForSecurityHub` service-linked role to be successfully created, the IAM identity that you use Security Hub with must have the required permissions. To grant the required permissions, attach the following policy to this IAM user, group, or role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "securityhub:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "securityhub.amazonaws.com"
        }
      }
    }
  ]
}
```



```
}  
  ]  
}
```

Creating a service-linked role for Security Hub

The `AWSServiceRoleForSecurityHub` service-linked role is automatically created when you enable Security Hub for the first time or enable Security Hub in a supported Region where you previously didn't have it enabled. You can also create the `AWSServiceRoleForSecurityHub` service-linked role manually using the IAM console, the IAM CLI, or the IAM API.

Important

The service-linked role that is created for the Security Hub administrator account doesn't apply to the Security Hub member accounts.

For more information about creating the role manually, see [Creating a service-linked role](#) in the *IAM User Guide*.

Editing a service-linked role for Security Hub

Security Hub doesn't allow you to edit the `AWSServiceRoleForSecurityHub` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role by using IAM. For more information, see [Editing a service-linked role](#) in the *IAM User Guide*.

Deleting a service-linked role for Security Hub

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete that role. That way, you don't have an unused entity that isn't actively monitored or maintained.

Important

To delete the `AWSServiceRoleForSecurityHub` service-linked role, you must first disable Security Hub in all Regions where it's enabled.

If Security Hub isn't disabled when you try to delete the service-linked role, the deletion fails. For more information, see [Disabling AWS Security Hub \(p. 616\)](#).

When you disable Security Hub, the `AWSServiceRoleForSecurityHub` service-linked role is *not* automatically deleted. If you enable Security Hub again, it starts using the existing `AWSServiceRoleForSecurityHub` service-linked role.

To manually delete the service-linked role using IAM

Use the IAM console, the IAM CLI, or the IAM API to delete the `AWSServiceRoleForSecurityHub` service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

AWS managed policies for AWS Security Hub

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to [create IAM customer managed policies](#) that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see [AWS managed policies](#) in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to

support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the `ViewOnlyAccess` AWS managed policy provides read-only access to many AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see [AWS managed policies for job functions](#) in the *IAM User Guide*.

AWS managed policy: `AWSSecurityHubFullAccess`

You can attach the `AWSSecurityHubFullAccess` policy to your IAM identities.

This policy grants administrative permissions that allow a principal full access to all Security Hub actions. This policy must be attached to a principal before they enable Security Hub manually for their account. For example, principals with these permissions can both view and update the status of findings. They can configure custom insights, and enable integrations. They can enable and disable standards and controls. Principals for an administrator account can also manage member accounts.

Permissions details

This policy includes the following permissions.

- `securityhub` – Allows principals full access to all Security Hub actions.
- `iam` – Allows principals to create a service-linked role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "securityhub:*",
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "iam:CreateServiceLinkedRole",
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "iam:AWSServiceName": "securityhub.amazonaws.com"
        }
      }
    }
  ]
}
```

Security Hub managed policy: `AWSSecurityHubReadOnlyAccess`

You can attach the `AWSSecurityHubReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions that allow users to view information in Security Hub. Principals with this policy attached cannot make any updates in Security Hub. For example, principals with these

permissions can view the list of findings associated with their account, but cannot change the status of a finding. They can view the results of insights, but cannot create or configure custom insights. They cannot configure controls or product integrations.

Permissions details

This policy includes the following permissions.

- `securityhub` – Allows users to perform actions that return either a list of items or details about an item. This includes API operations that start with `Get`, `List`, or `Describe`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "securityhub:Get*",
        "securityhub:List*",
        "securityhub:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

AWS managed policy: AWSSecurityHubOrganizationsAccess

You can attach the `AWSSecurityHubOrganizationsAccess` policy to your IAM identities.

This policy grants administrative permissions in AWS Organizations that are required to support the Security Hub integration with Organizations.

These permissions allow the organization management account to designate the delegated administrator account for Security Hub. They also allow the delegated Security Hub administrator account to enable organization accounts as member accounts.

This policy only provides the permissions for Organizations. The organization management account and delegated Security Hub administrator account also require permissions for the associated actions in Security Hub. These permissions can be granted using the `AWSSecurityHubFullAccess` managed policy.

Permissions details

This policy includes the following permissions.

- `organizations:ListAccounts` – Allows principals to retrieve the list of accounts that belong to an organization.
- `organizations:DescribeOrganization` – Allows principals to retrieve information about the organization configuration.
- `organizations:EnableAWSServiceAccess` – Allows principals to enable the Security Hub integration with Organizations.
- `organizations:RegisterDelegatedAdministrator` – Allows principals to designate the delegated administrator account for Security Hub.
- `organizations:DeregisterDelegatedAdministrator` – Allows principals to remove the delegated administrator account for Security Hub.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "organizations:ListAccounts",
        "organizations:DescribeOrganization"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": "organizations:EnableAWSServiceAccess",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "organizations:ServicePrincipal": "securityhub.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "organizations:RegisterDelegatedAdministrator",
        "organizations:DeregisterDelegatedAdministrator"
      ],
      "Resource": "arn:aws:organizations::*:account/o-*/*",
      "Condition": {
        "StringEquals": {
          "organizations:ServicePrincipal": "securityhub.amazonaws.com"
        }
      }
    }
  ]
}
```

AWS managed policy: AWSSecurityHubServiceRolePolicy

You can't attach `AWSSecurityHubServiceRolePolicy` to your IAM entities. This policy is attached to a service-linked role that allows Security Hub to perform actions on your behalf. For more information, see [the section called "Using service-linked roles" \(p. 80\)](#).

This policy grants administrative permissions that allow the service-linked role to perform the security checks for Security Hub controls.

Permissions details

This policy includes permissions to do the following:

- `cloudtrail` – Retrieve information about CloudTrail trails.
- `cloudwatch` – Retrieve the current CloudWatch alarms.
- `logs` – Retrieve the metric filters for CloudWatch logs.
- `sns` – Retrieve the list of subscriptions to an SNS topic.
- `config` – Retrieve information about configuration recorders, resources, and AWS Config rules. Also allows the service-linked role to create and delete AWS Config rules, and to run evaluations against the rules.
- `iam` – Get and generate credential reports for accounts.

- `organizations` – Retrieve account information for an organization.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudtrail:DescribeTrails",
        "cloudtrail:GetTrailStatus",
        "cloudtrail:GetEventSelectors",
        "cloudwatch:DescribeAlarms",
        "logs:DescribeMetricFilters",
        "sns:ListSubscriptionsByTopic",
        "config:DescribeConfigurationRecorders",
        "config:DescribeConfigurationRecorderStatus",
        "config:DescribeConfigRules",
        "config:BatchGetResourceConfig",
        "config:PutEvaluations",
        "config:SelectResourceConfig",
        "iam:GenerateCredentialReport",
        "iam:GetCredentialReport",
        "organizations:ListAccounts",
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "config:PutConfigRule",
        "config>DeleteConfigRule",
        "config:GetComplianceDetailsByConfigRule",
        "config:DescribeConfigRuleEvaluationStatus"
      ],
      "Resource": "arn:aws:config:*:*:config-rule/aws-service-rule/*securityhub*"
    }
  ]
}
```

Security Hub updates to AWS managed policies

View details about updates to AWS managed policies for Security Hub since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Security Hub [Document history \(p. 618\)](#) page.

Change	Description	Date
AWSSecurityHubServiceRolePolicy – Update to an existing policy	<p>Security Hub moved the existing <code>config:PutEvaluations</code> permission to a different statement within the policy.</p> <p>The <code>config:PutEvaluations</code> permission is now applied to all resources.</p>	July 14, 2021

Change	Description	Date
AWSSecurityHubServiceRolePolicy – Update to an existing policy	Security Hub added a new permission to allow the service-linked role to deliver evaluation results to AWS Config.	June 29, 2021
AWSSecurityHubServiceRolePolicy – Added to the list of managed policies	Added information about the managed policy AWSSecurityHubServiceRolePolicy , which is used by the Security Hub service-linked role.	June 11, 2021
AWSSecurityHubOrganizations – New policy	Security Hub added a new policy that grants permissions that are needed for the Security Hub integration with Organizations.	March 15, 2021
Security Hub started tracking changes	Security Hub started tracking changes for its AWS managed policies.	March 15, 2021

Compliance validation for AWS Security Hub

Third-party auditors assess the security and compliance of AWS Security Hub as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

For a list of AWS services in scope of specific compliance programs, see [AWS Services in Scope by Compliance Program](#). For general information, see [AWS Compliance Programs](#).

You can download third-party audit reports using AWS Artifact. For more information, see [Downloading reports in AWS Artifact](#).

Your compliance responsibility when using Security Hub is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- [Security and Compliance Quick Start Guides](#) – These deployment guides discuss architectural considerations and provide steps for deploying security- and compliance-focused baseline environments on AWS.
- [AWS Compliance Resources](#) – This collection of workbooks and guides might apply to your industry and location.
- [AWS Config](#) – This AWS service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- [AWS Security Hub](#) – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices.

Infrastructure security in AWS Security Hub

As a managed service, AWS Security Hub is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

You use AWS published API calls to access Security Hub through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support

cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the [AWS Security Token Service](#) (AWS STS) to generate temporary security credentials to sign requests.

AWS Security Hub and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and AWS Security Hub by creating an *interface VPC endpoint*. Interface endpoints are powered by [AWS PrivateLink](#), a technology that enables you to privately access Security Hub APIs without an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to communicate with Security Hub APIs. Traffic between your VPC and Security Hub does not leave the Amazon network.

Each interface endpoint is represented by one or more [Elastic Network Interfaces](#) in your subnets.

For more information, see [Interface VPC endpoints \(AWS PrivateLink\)](#) in the *AWS PrivateLink Guide*.

Considerations for Security Hub VPC endpoints

Before you set up an interface VPC endpoint for Security Hub, ensure that you review [Interface endpoint properties and limitations](#) in the *AWS PrivateLink Guide*.

Security Hub supports making calls to all of its API actions from your VPC.

Note

Security Hub does not support VPC endpoints in the Asia Pacific (Osaka) Region.

Creating an interface VPC endpoint for Security Hub

You can create a VPC endpoint for the Security Hub service using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see [Create an interface endpoint](#) in the *AWS PrivateLink Guide*.

Create a VPC endpoint for Security Hub using the following service name:

- `com.amazonaws.region.securityhub`

If you enable private DNS for the endpoint, you can make API requests to Security Hub using its default DNS name for the Region, for example, `securityhub.us-east-1.amazonaws.com`.

For more information, see [Access a service through an interface endpoint](#) in the *AWS PrivateLink Guide*.

Creating a VPC endpoint policy for Security Hub

You can attach an endpoint policy to your VPC endpoint that controls access to Security Hub. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see [Control access to services with VPC endpoints](#) in the *AWS PrivateLink Guide*.

Example: VPC endpoint policy for Security Hub actions

The following is an example of an endpoint policy for Security Hub. When attached to an endpoint, this policy grants access to the listed Security Hub actions for all principals on all resources.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "securityhub:getFindings",
        "securityhub:getEnabledStandards",
        "securityhub:getInsights"
      ],
      "Resource": "*"
    }
  ]
}
```


Managing administrator and member accounts

An administrator account can view data from and manage configuration for its member accounts. The administrator-member relationship is established differently based on whether you use the integration with AWS Organizations.

If you are integrated with Organizations, the organization management account designates the Security Hub administrator account. See [the section called “Designating a Security Hub administrator account” \(p. 93\)](#). The Security Hub administrator account automatically has access to all of the accounts in the organization. The Security Hub administrator account determines which organization accounts to enable as member accounts. See [the section called “Managing organization member accounts” \(p. 98\)](#). These member accounts cannot disassociate themselves from the administrator account.

Otherwise, member accounts accept an invitation from an administrator account. The Security Hub administrator account can also invite member accounts that are not part of the organization. See [the section called “Managing member accounts by invitation” \(p. 102\)](#). Accounts that are added by invitation can disassociate themselves from their administrator account.

Topics

- [Effects of an administrator-member relationship \(p. 89\)](#)
- [Restrictions and recommendations \(p. 90\)](#)
- [Making the transition to AWS Organizations for account management \(p. 91\)](#)
- [Allowed actions for accounts \(p. 92\)](#)
- [Designating a Security Hub administrator account \(p. 93\)](#)
- [Managing member accounts that belong to an organization \(p. 98\)](#)
- [Managing member accounts by invitation \(p. 102\)](#)
- [Effect of account actions on Security Hub data \(p. 108\)](#)

Effects of an administrator-member relationship

An administrator account is granted permission to view the findings that are associated with their member accounts. This also allows the administrator account to view insight results and control statuses from across their member accounts. Administrator accounts can also take action on their member accounts' findings.

Security Hub does not copy member account findings into the administrator account. Administrator accounts also cannot change the Security Hub configuration for member accounts. See [the section called “Allowed actions for accounts” \(p. 92\)](#).

In Security Hub, all findings are ingested into a specific Region for a specific account.

In each Region, the administrator account can view and manage findings for their member accounts in that Region.

In an aggregation Region, the administrator account can view and manage member account findings from linked Regions that are replicated to the aggregation Region. For more information about cross-Region aggregation, see [Cross-Region aggregation \(p. 111\)](#).

Restrictions and recommendations

Maximum number of member accounts

Security Hub supports up to 5,000 member accounts per administrator account in each Region.

Accounts and Regions

Membership by organization

If you are enrolled in AWS Organizations, the organization management account can designate a Security Hub administrator account in each Region.

The Security Hub administrator account for a Region also becomes that Region's delegated administrator account for Security Hub in Organizations. The exception is if the organization management account designates itself as the Security Hub administrator account. The organization management account cannot be a delegated administrator in Organizations.

Once the delegated administrator account for a Region is set in Organizations, the organization management account can choose either the delegated administrator account or itself as the Security Hub administrator account in that Region. We recommend that you choose the same delegated administrator account in all Regions.

The Security Hub administrator account manages member accounts separately in each Region.

Membership by invitation

For member accounts created by invitation, the administrator-member account association is created only in the Region that the invitation is sent from. The administrator account must enable Security Hub in each Region that you want to use it in. The administrator account then invites each account to associate as a member account in that Region.

Restrictions on administrator-member relationships

An account cannot be an administrator account and a member account at the same time.

A member account can only be associated with one administrator account. If an organization account is enabled by the Security Hub administrator account, the account cannot accept an invitation from another account. If an account has accepted an invitation, the account cannot be enabled by the Security Hub administrator account for the organization. It also cannot receive invitations from other accounts.

For the manual invitation process, accepting a membership invitation is optional.

Coordinating administrator accounts across services

Security Hub aggregates findings from various AWS services, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie. Security Hub also allows users to pivot from a GuardDuty finding to start an investigation in Amazon Detective.

However, the administrator-member relationships that you set up in these other services do not automatically apply to Security Hub. Security Hub recommends that you use the same account as the administrator account for all of these services. This administrator account should be an account that is responsible for security tools. The same account should also be the aggregator account for AWS Config.

For example, a user from the GuardDuty administrator account A can see findings for GuardDuty member accounts B and C on the GuardDuty console. If account A then enables Security Hub, users from

account A do *not* automatically see GuardDuty findings for accounts B and C in Security Hub. A Security Hub administrator-member relationship is also required for these accounts.

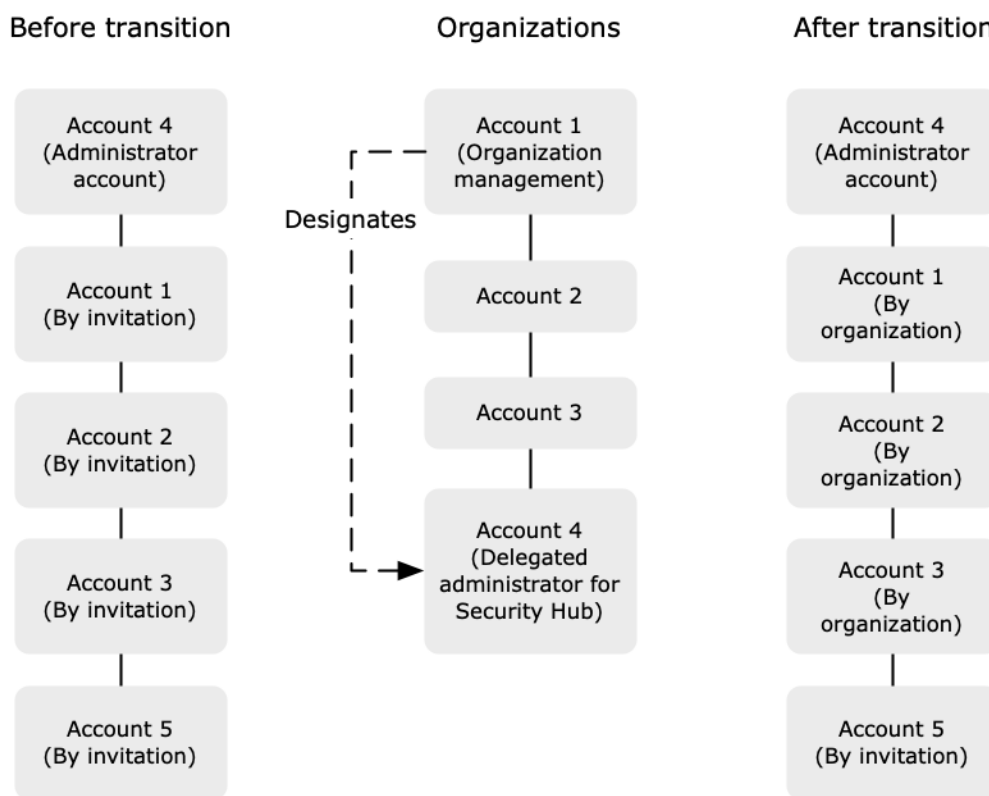
To do this, make account A the Security Hub administrator account and enable accounts B and C to become Security Hub member accounts.

Making the transition to AWS Organizations for account management

You might have an existing administrator account with member accounts that accepted a manual invitation. If you are enrolled in AWS Organizations, use the following steps to use Organizations to enable and manage member accounts instead of using the manual invitation process:

1. [Designate a Security Hub administrator account for your organization. \(p. 91\)](#)
2. [Enable organization accounts under the Security Hub administrator account. Existing member accounts are enabled automatically. \(p. 92\)](#)

The following diagram shows an overview of the administrator and member account structure before the transition, the configuration in Organizations, and the account structure after the transition.



Designate a Security Hub administrator account for your organization

Your organization management account designates the Security Hub administrator account for your organization. See [the section called "Designating a Security Hub administrator account" \(p. 93\)](#). The

Security Hub administrator account also becomes the delegated administrator account for Security Hub in Organizations.

To make the transition simpler, Security Hub recommends that you choose the current administrator account as the Security Hub administrator account for the organization. This is because a member account cannot belong to more than one administrator account. The administrator account for the organization cannot enable any organization accounts that are member accounts under another administrator account.

Enable organization accounts as member accounts

The Security Hub administrator account determines which organization accounts to enable as member accounts. See [the section called “Managing organization member accounts” \(p. 98\)](#).

On the **Accounts** page, the Security Hub administrator account sees all of the accounts in the organization. Organization accounts have a type of **By organization**, even if they were previously member accounts by invitation.

If the Security Hub administrator account was already an administrator account, all of their existing member accounts are enabled as member accounts automatically. Existing member accounts that are not organization accounts have a type of **By invitation**.

The **Accounts** page also provides an option to automatically enable new accounts as they are added to an organization. See [the section called “Enabling new accounts automatically” \(p. 99\)](#). The option is initially turned off (**Auto-enable is off**).

Until you enable that option, the **Accounts** page displays a message that contains an **Enable** button. When you choose **Enable**, Security Hub performs the following actions:

- Enables all of the organization accounts as member accounts, except for accounts that are member accounts under another administrator account.

Before the Security Hub administrator account can enable those accounts, they must be disassociated from the other administrator account. See [the section called “Disassociating member accounts” \(p. 106\)](#).

If an organization account does not have Security Hub enabled, then Security Hub and the default standards are enabled automatically for that account.

For organization accounts that already have Security Hub enabled, Security Hub does not make any other changes to the account. It only enables the membership.

- Toggles the setting to enable new accounts automatically (**Auto-enable is on**).

Allowed actions for accounts

Administrator and member accounts have access to the following Security Hub actions. In the table, the values have the following meanings:

- **Any** – The account can perform the action for all of the accounts under the same administrator or account.
- **Self** – The account can only perform the action on their own account.

A dash (–) indicates that the account cannot perform the action.

This table reflects the default permissions for administrator and member accounts. You can use custom IAM policies to further restrict access to Security Hub features and functions. For guidance and examples, see the blog post [Aligning IAM policies to user personas for AWS Security Hub](#).

Action	Security Hub administrator account (Organization)	Administrator account (Invitation)	Member (Organization)	Member (Invitation)
View accounts	Any	Any	-	-
Disassociate member account	Any	Any	-	Self
Delete member account	Any non-organization account	Any	-	-
Disable Security Hub	-	Self - if there are no enabled member accounts	Self - if disassociated from the Security Hub administrator account	Self - if disassociated from the administrator account
View findings	Any	Any	Self	Self
Update findings	Any	Any	Self	Self
View insight results	Any	Any	Self	Self
View control details	Any	Any	Self	Self
Enable and disable controls	Self	Self	Self	Self
Enable and disable standards	Self	Self	Self	Self
Enable and disable integrations	Self	Self	Self	Self
Configure custom actions	Self	Self	Self	Self

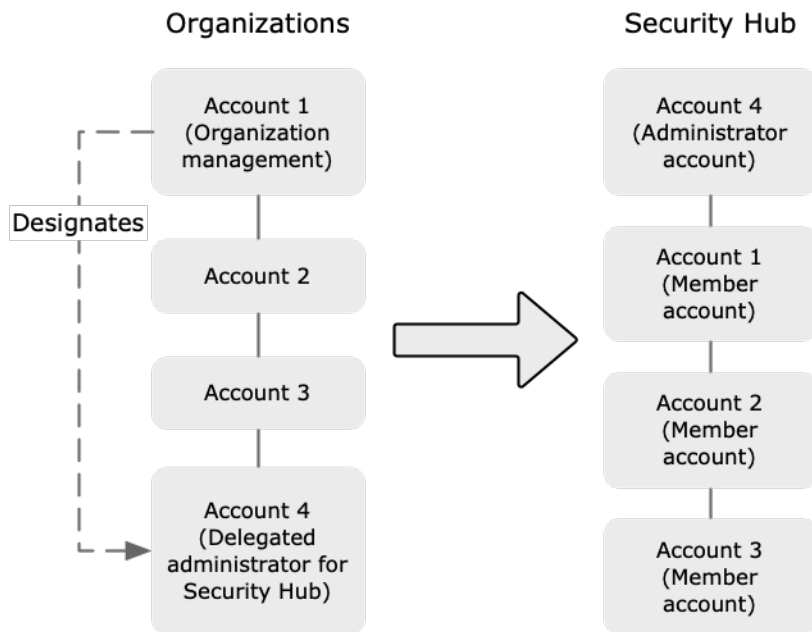
Designating a Security Hub administrator account

The Security Hub administrator account manages Security Hub membership for an organization.

How the Security Hub administrator account is managed

The organization management account designates the Security Hub administrator account in each Region.

The Security Hub administrator account then enables organization accounts as member accounts. They can also invite other accounts to be member accounts. See [the section called "Managing organization member accounts" \(p. 98\)](#) and [the section called "Managing member accounts by invitation" \(p. 102\)](#).



Member accounts can only be associated with a single administrator account. The Security Hub administrator account cannot enable member accounts that belong to another administrator account.

All Security Hub accounts must have AWS Config enabled and configured to record all resources. For details on the requirement for AWS Config, see [the section called “Enabling AWS Config” \(p. 8\)](#).

Setting the Security Hub administrator account as the delegated administrator account

When you first choose a Security Hub administrator account, Security Hub calls Organizations to make that account the delegated administrator account for Security Hub.

Once you have a delegated administrator account in Organizations, then you can choose either that account or the organization management account as the Security Hub administrator account in all Regions. We recommend choosing the same delegated administrator account in all Regions.

To choose a different account, you must remove the current Security Hub administrator account in all Regions.

Recommendations for choosing the Security Hub administrator account

If you have an administrator account in place from the manual invitation process, then Security Hub recommends that you designate that account as the Security Hub administrator account.

We also recommend that you do not designate the organization management account itself as the Security Hub administrator account. This is because the users who have access to the organization management account to manage billing are likely to be different from the users who need access to Security Hub for security management.

The organization management account also cannot be the delegated administrator account for a service in Organizations.

Removing the Security Hub administrator account

The organization management account can remove the Security Hub administrator account.

When the organization management account uses the console to remove the Security Hub administrator account in one Region, it is automatically removed in all Regions. Security Hub also calls Organizations to remove the delegated administrator account.

The Security Hub API only removes the Security Hub administrator account from the Region where the API call or command is issued. It does not update other Regions, and it does not remove the delegated administrator account in Organizations.

When you use the Organizations API to remove the delegated administrator account for Security Hub, Security Hub also removes the Security Hub administrator account in all Regions.

Required permissions to configure the Security Hub administrator account

To designate and remove a Security Hub administrator account, the organization management account must have permissions for the `EnableOrganizationAdminAccount` and `DisableOrganizationAdminAccount` actions in Security Hub. The organization management account must also have administrative permissions for Organizations.

To grant all of the required permissions, attach the following Security Hub managed policies to the IAM principal for the organization management account:

- [AWSSecurityHubFullAccess \(p. 82\)](#)
- [AWSSecurityHubOrganizationsAccess \(p. 83\)](#)

Designating a Security Hub administrator account (console)

The organization management account can use the Security Hub console to designate the Security Hub administrator account.

The organization management account does not have to enable Security Hub in order to manage the Security Hub administrator account.

Security Hub recommends that the organization management account is not the Security Hub administrator account. However, if the organization management account does choose itself as the Security Hub administrator account, it must have Security Hub enabled. If it does not have Security Hub enabled, it must enable Security Hub manually. Security Hub cannot be enabled automatically for the organization management account.

To designate a Security Hub administrator account from the Welcome to Security Hub page

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Choose **Go to Security Hub**.
3. If a Security Hub administrator account is currently assigned, then you must remove the current account before you can designate a new account.

To remove the current account, under **Delegated Administrator**, choose **Remove**.

4. Under **Delegated Administrator**, enter the account ID of the account to designate as the **Security Hub** administrator account.

You must designate the same Security Hub administrator account in all Regions. If you designate an account that is different from the account designated in other Regions, Security Hub returns an error.

5. Choose **Delegate**.

If you have Security Hub enabled, then you can also designate the Security Hub administrator account from the **Settings** page.

To designate a Security Hub administrator account from the Settings page

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the Security Hub navigation pane, choose **Settings**. Then choose **General**.
3. If a Security Hub administrator account is currently assigned, then before you can designate a new account, you must remove the current account.

Under **Delegated Administrator**, to remove the current account, choose **Remove**.

4. Enter the account ID of the account you want to designate as the **Security Hub** administrator account.

You must designate the same Security Hub administrator account in all Regions. If you designate an account that is different from the account designated in other Regions, Security Hub returns an error.

5. Choose **Delegate**.

Designating a Security Hub administrator account (Security Hub API, AWS CLI)

To designate the Security Hub administrator account, you can use an API call or the AWS Command Line Interface. You must use the organization management account credentials.

To designate the **Security Hub** administrator account (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [EnableOrganizationAdminAccount](#) operation. You must provide the AWS account ID of the Security Hub administrator account.
- **AWS CLI** – At the command line, run the `enable-organization-admin-account` command.

```
aws securityhub enable-organization-admin-account --admin-account-id <admin account ID>
```

Example

```
aws securityhub enable-organization-admin-account --admin-account-id 777788889999
```

Removing a Security Hub administrator account (console)

The organization management account can remove the current Security Hub administrator account. When you use the console to remove the Security Hub administrator account, the Security Hub administrator account is removed in all Regions. Security Hub also calls Organizations to remove the delegated administrator account for Security Hub.

When the Security Hub administrator account is removed, the member accounts are disassociated from the removed Security Hub administrator account.

The enabled member accounts still have Security Hub enabled. They become standalone accounts until a new Security Hub administrator enables them as member accounts.

If the organization management account is not an enabled account in Security Hub, then use the option on the **Welcome to Security Hub** page.

To remove the Security Hub administrator account from the Welcome to Security Hub page

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Choose **Go to Security Hub**.
3. Under **Delegated Administrator**, choose **Remove**.

If the organization management account is an enabled account in **Security Hub**, then use the option on the **General** tab of the **Settings** page.

To remove the Security Hub administrator account from the Settings page

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the Security Hub navigation pane, choose **Settings**. Then choose **General**.
3. Under **Delegated Administrator**, choose **Remove**.

Removing a Security Hub administrator account (Security Hub API, AWS CLI)

To remove the Security Hub administrator account, you can use an API call or the AWS Command Line Interface. You must use the organization management account credentials.

When you use the API or AWS CLI to remove the Security Hub administrator account, it is only removed in the Region where the API call or command was issued. Security Hub does not update other Regions, and it does not remove the delegated administrator account in Organizations.

To remove the Security Hub administrator account (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisableOrganizationAdminAccount` operation. You must provide the account ID of the Security Hub administrator account.
- **AWS CLI** – At the command line, run the `disable-organization-admin-account` command.

```
aws securityhub disable-organization-admin-account --admin-account-id <admin account ID>
```

Example

```
aws securityhub disable-organization-admin-account --admin-account-id 777788889999
```

Removing the delegated administrator account (Organizations API, AWS CLI)

When you use the Security Hub API to remove the Security Hub administrator account, it is only removed in the Region where the API call or command was issued. Security Hub does not update other Regions, and does not remove the delegated administrator account in Organizations.

The Organizations API allows you to remove the delegated administrator account. When you remove the delegated administrator account for Security Hub, Security Hub also removes the Security Hub administrator account from all Regions.

To remove the delegated administrator account (Organizations API, AWS CLI)

- **Organizations API** – Use the [DeregisterDelegatedAdministrator](#) operation. You must provide the account ID of the delegated administrator account, and the service principal for Security Hub, which is `securityhub.amazonaws.com`.
- **AWS CLI** – At the command line, run the `deregister-delegated-administrator` command.

```
aws organizations deregister-delegated-administrator --account-id <admin account ID> --  
service-principal <Security Hub service principal>
```

Example

```
aws organizations deregister-delegated-administrator --account-id 777788889999 --service-  
principal securityhub.amazonaws.com
```

Managing member accounts that belong to an organization

For organization accounts, the Security Hub administrator account can perform the following actions.

- Automatically treat *new* organization accounts as Security Hub member accounts as they are added to the organization.
- Treat *existing* organization accounts as Security Hub member accounts.
- Disassociate and delete member accounts that belong to the organization.

To ensure that the Security Hub administrator account has the required permissions to manage the organization accounts, attach the following managed policies to the associated IAM principal.

- [AWSSecurityHubFullAccess](#) (p. 82)
- [AWSSecurityHubOrganizationsAccess](#) (p. 83)

If the Security Hub administrator account has not turned on the option to automatically enable new organization accounts, then the **Accounts** page displays a message at the top of the page. The message contains an **Enable** option.

Note

Choosing **Enable** from this message impacts newly added and existing organization accounts. Turning on **Auto-enable accounts** only impacts newly added organization accounts. For more information, see [Automatically enabling new organization accounts](#) (p. 99).

When you choose **Enable**, Security Hub treats all of the existing organization accounts as member accounts, and automatically treats new accounts as member accounts as they are added to the organization.

For organization accounts that do not have Security Hub enabled, enables Security Hub, and enables the CIS AWS Foundations Benchmark standard and the AWS Foundational Best Practices standard. For organization accounts that already have Security Hub enabled, Security Hub does not make any other changes to those accounts. It does not change their enabled standards or controls.

Topics

- [Automatically enabling new organization accounts \(p. 99\)](#)
- [Enabling member accounts from your organization \(p. 100\)](#)
- [Disassociating member accounts from your organization \(p. 101\)](#)

Automatically enabling new organization accounts

The Security Hub administrator account can configure Security Hub to automatically enable new organization accounts as member accounts.

When new accounts are added to your organization, they are added to the list on the **Accounts** page. For organization accounts, **Type** is **By organization**. By default, the new accounts are not enabled as member accounts. Their status is **Not a member**.

When you turn on automatic enablement, Security Hub treats *new* accounts as member accounts when they are added to the organization. Turning on automatic enablement does not treat *existing* organization accounts as member accounts unless they were already enabled as member accounts. Security Hub also cannot automatically treat accounts as members if they already belong to another administrator account.

If an organization account does not have Security Hub enabled, then Security Hub and the [default standards \(p. 334\)](#) are enabled automatically for that account.

For organization accounts that already have Security Hub enabled, Security Hub does not make any other changes to the account. It only creates the membership.

Remember that all Security Hub accounts must have AWS Config enabled and configured to record all resources. For details on the requirement for AWS Config, see [the section called "Enabling AWS Config" \(p. 8\)](#).

Enabling Security Hub automatically for new accounts (console)

The **Accounts** page includes a configuration option to automatically add new accounts.

To automatically enable new organization accounts as member accounts

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the Security Hub navigation pane, choose **Settings**.
3. On the **Settings** page, choose **Accounts**.
4. On the **Accounts** tab, toggle the automatic enablement setting to **Auto-enable is on**.

Enabling Security Hub automatically for new organization accounts (Security Hub API, AWS CLI)

To determine whether to automatically enable new organization accounts, the Security Hub administrator account can use the Security Hub API or the AWS Command Line Interface.

To automatically enable new organization accounts

- **Security Hub API** – Use the [UpdateOrganizationConfiguration](#) operation. To automatically enable new organization accounts, set `autoEnable` to `true`.
- **AWS CLI** – At the command line, run the `update-organization-configuration` command.

```
aws securityhub update-organization-configuration --auto-enable
```

Enabling member accounts from your organization

If you do not automatically enable new organization accounts as member accounts, then you can enable those accounts manually. You must also manually enable accounts that you disassociated.

You cannot enable an account if it is already a member account for a different administrator account.

You also cannot enable an account that is currently suspended. If you try to enable a suspended account, the account status changes to **Account Suspended**.

When you enable an organization account as a member account, the following occurs:

- If the account does not have Security Hub enabled, Security Hub is enabled for that account. The AWS Foundational Security Best Practices and CIS AWS Foundations Benchmark standards also are enabled for the account. The account does not receive an invitation.

The exception to this is the organization management account. Security Hub cannot be enabled automatically for the organization management account. The organization management account must enable Security Hub before you enable the organization management account as a member account.

- If the account already has Security Hub enabled, Security Hub does not make any other changes to the account. It only enables the membership.

Remember that all Security Hub accounts must have AWS Config enabled and configured to record all resources. For details on the requirement for AWS Config, see [the section called “Enabling AWS Config” \(p. 8\)](#).

Enabling an organization account as a member account (console)

In the **Accounts** list, an organization account that was either never enabled or that was disassociated from the Security Hub administrator account has a status of **Not a member**.

To enable an organization account as a member account

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the Security Hub navigation pane, choose **Settings**. Then choose **Accounts**.
3. In the **Accounts** list, select the check box for each organization account that you want to enable.
4. Choose **Actions**, then choose **Add member**.

Enabling an organization account as a member account (Security Hub API, AWS CLI)

The Security Hub administrator account can use the Security Hub API or AWS Command Line Interface to enable organization accounts. Unlike the manual invitation process, when you use [CreateMembers](#) to enable an organization account, you do not need to send an invitation.

To enable organization accounts as member accounts

- **Security Hub API** – Use the `CreateMembers` operation. For each account to enable, you provide the account ID.
- **AWS CLI** – At the command line, run the `create-members` command.

```
aws securityhub create-members --account-details '[{"AccountId": "<accountId>"}]'
```

Example

```
aws securityhub create-members --account-details '[{"AccountId": "123456789111"}, {"AccountId": "123456789222"}]'
```

Disassociating member accounts from your organization

To stop receiving and viewing findings from an enabled member account, you can disassociate the member account.

When you disassociate a member account, the status changes to **Not a member**.

Disassociating member accounts (console)

To disassociate member accounts

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**. Then choose **Accounts**.
3. In the **Accounts** list, select the accounts to disassociate. You can only disassociate **Enabled** accounts.
4. Choose **Actions**, and then choose **Disassociate account**.

Disassociating an account (Security Hub API, AWS CLI)

To disassociate member accounts, you can use the Security Hub API or the AWS Command Line Interface.

To disassociate member accounts (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisassociateMembers` operation. You must provide the AWS account IDs for the member accounts to disassociate. To view a list of member accounts, use the `ListMembers` operation.
- **AWS CLI** – At the command line, run the `disassociate-members` command.

```
aws securityhub disassociate-members --account-ids "<accountIds>"
```

Example

```
aws securityhub disassociate-members --account-ids "123456789111" "123456789222"
```

Managing member accounts by invitation

AWS Security Hub also supports a manual invitation process. You use the manual process if you do not use AWS Organizations.

You also use this process for accounts that do not belong to your organization. For example, you might not include a test account in your organization. Or you might want to consolidate accounts from multiple organizations under a single Security Hub administrator account. The Security Hub administrator account must send invitations to accounts that belong to other organizations.

On the **Accounts** tab of the **Settings** page, accounts that were added by invitation have **Type** set to **By invitation**.

If you do not use Organizations at all, then an account becomes an administrator account when a member account accepts an invitation.

Topics

- [Adding and inviting member accounts \(p. 102\)](#)
- [Responding to an invitation to be a member account \(p. 104\)](#)
- [Disassociating member accounts \(p. 106\)](#)
- [Deleting member accounts \(p. 107\)](#)
- [Disassociating from your administrator account \(p. 107\)](#)

Adding and inviting member accounts

Your account becomes the administrator account for accounts that accept your invitation.

When you accept an invitation from another account, your account becomes a member account, and that account becomes your administrator account.

If your account is an administrator account, you cannot accept an invitation to become a member account.

Adding a member account consists of the following steps:

1. The administrator account adds the member account to their list of member accounts.
2. The administrator account sends an invitation to the member account.
3. The member account accepts the invitation.

Adding member accounts (console)

From the Security Hub console, you can add accounts to your list of member accounts. You can select accounts individually, or upload a `.csv` file that contains the account information.

For each account, you must provide the account ID and an email address. The email address should be the email address to contact about security issues in the account. It is not used to verify the account.

To add accounts to your list of member accounts

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the left pane, choose **Settings**.
3. On the **Settings** page, choose **Accounts** and then choose **Add accounts**. You can then either add accounts individually or upload a `.csv` file containing the list of accounts.

- To select the accounts, do one of the following:
 - To add the accounts individually, under **Enter accounts**, enter the account ID and email address of the account to add, and then choose **Add**.

Repeat this process for each account.

- To use a comma-separated values (.csv) file to add multiple accounts, first create the file. The file must contain the account ID and email address for each account to add.

In your .csv list, accounts must appear one per line. The first line of the .csv file must contain the header. In the header, the first column is **Account ID** and the second column is **Email**.

Each subsequent line must contain a valid account ID and email address for the account to add.

Here is an example of a .csv file when viewed in a text editor.

```
Account ID,Email
111111111111,user@example.com
```

In a spreadsheet program, the fields appear in separate columns. The underlying format is still comma-separated. You must format the account IDs as non-decimal numbers. For example, the account ID 444455556666 cannot be formatted as 444455556666.0. Also make sure that the number formatting does not remove any leading zeros from the account ID.

To select the file, on the console, choose **Upload list (.csv)**. Then choose **Browse**.

After you select the file, choose **Add accounts**.

- After you finish adding accounts, under **Accounts to be added**, choose **Next**.

Inviting member accounts (console)

After you add the member accounts, you send an invitation to the member account. You can also resend an invitation to an account that you disassociated.

To send an invitation to a new account

- Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
- In the navigation pane, choose **Settings**, and then choose **Accounts**.
- For the account to invite, choose **Invite** in the **Status** column.
- When prompted to confirm, choose **Invite**.

To resend an invitation to accounts that you disassociated

- Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
- In the navigation pane, choose **Settings**, and then choose **Accounts**.
- Select each disassociated account that you want to resend an invitation to.
- From **Actions**, choose **Resend invitation**.

Adding member accounts (Security Hub API, AWS CLI)

To add member accounts, you can use an API call or the AWS Command Line Interface. You must use the administrator account credentials. Only the administrator account can perform this action.

To add member accounts (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [CreateMembers](#) operation. For each member account to add, you must provide the AWS account ID.
- **AWS CLI** – At the command line, run the `create-members` command.

```
aws securityhub create-members --account-details '[{"AccountId": "<accountID1>"}]'
```

Example

```
aws securityhub create-members --account-details '[{"AccountId": "123456789111"}, {"AccountId": "123456789222"}]'
```

Inviting member accounts (Security Hub API, AWS CLI)

To invite accounts that you added, you can use an API call or the AWS Command Line Interface. You use the same API operation or AWS CLI command to resend invitations to member accounts that you disassociated. You must use the administrator account credentials. Only the administrator account can perform this action.

To invite member accounts (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [InviteMembers](#) operation. For each account to invite, you must provide the AWS account ID.
- **AWS CLI** – At the command line, run the `invite-members` command.

```
aws securityhub invite-members --account-ids <accountIDs>
```

Example

```
aws securityhub invite-members --account-ids "123456789111" "123456789222"
```

Responding to an invitation to be a member account

You can accept or decline an invitation to be a member account.

After you accept an invitation, your account becomes an AWS Security Hub member account. The account that sent the invitation becomes your Security Hub administrator account. The administrator account user can view findings for your member account in Security Hub.

If you decline the invitation, then your account is marked as **Resigned** on the administrator account's list of member accounts.

You can only accept one invitation to be a member account.

Before you can accept or decline an invitation, you must enable Security Hub. For information on how to enable Security Hub, see [the section called “Enabling Security Hub manually” \(p. 66\)](#).

Remember that all Security Hub accounts must have AWS Config enabled and configured to record all resources. For details on the requirement for AWS Config, see [the section called “Enabling AWS Config” \(p. 8\)](#).

Accepting an invitation (console)

On the **Accounts** page, **Administrator account** contains the invitation and membership information for an account.

To accept an invitation to be a member account

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. Under **Administrator account**, toggle **Accept** to the on position, and then choose **Accept invitation**.

Accepting an invitation (Security Hub API, AWS CLI)

To accept an invitation to be a member account, you can use an API call or the AWS Command Line Interface. You must use the credentials for the member account that received the invitation.

To accept an invitation (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `AcceptAdministratorInvitation` operation. You must provide the invitation identifier and the AWS account ID of the administrator account. To retrieve details about the invitation, use the `ListInvitations` operation.
- **AWS CLI** – At the command line, run the `accept-administrator-invitation` command.

```
aws securityhub accept-administrator-invitation --administrator-id <administratorAccountID> --invitation-id <invitationID>
```

Example

```
aws securityhub accept-administrator-invitation --administrator-id 123456789012 --invitation-id 7ab938c5d52d7904ad09f9e7c20cc4eb
```

Note

The Security Hub console continues to use `AcceptInvitation`. It will eventually change to use `AcceptAdministratorInvitation`. Any IAM policies that specifically control access to this function must continue to use `AcceptInvitation`. You should also add `AcceptAdministratorInvitation` to your policies to ensure that the correct permissions are in place after the console begins to use `AcceptAdministratorInvitation`.

Declining an invitation (console)

You can decline an invitation to be a member account. When you decline an invitation, your account is marked as **Resigned** on the administrator account's list of member accounts.

To decline an invitation to be a member account

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. Under **Administrator account**, choose **Decline invitation**.

Declining an invitation (Security Hub API, AWS CLI)

To decline an invitation, you can use an API call or the AWS Command Line Interface.

To decline an invitation (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DeclineInvitations` operation. You must provide the AWS account ID of the administrator account that issued the invitation. To view information about your invitations, use the `ListInvitations` operation.
- **AWS CLI** – At the command line, run the `decline-invitations` command.

```
aws securityhub decline-invitations --account-ids "<administratorAccountId>"
```

Example

```
aws securityhub decline-invitations --account-ids "123456789012"
```

Disassociating member accounts

When you're logged into an administrator account, you can disassociate a member account to stop receiving and viewing findings from that account. You must disassociate a member account before you can delete it.

When you disassociate a member account, it remains in your list of member accounts with a status of **Removed (Disassociated)**. Your account is removed from the administrator account information for the member account.

To resume receiving findings for the account, you can resend the invitation. To remove the member account entirely, you can delete the member account.

Disassociating member accounts (console)

From the **Accounts** page, you can disassociate one or more member accounts.

To disassociate member accounts

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. Under **Member accounts**, select the accounts to disassociate.
4. Choose **Actions**, and then choose **Disassociate accounts**.

Disassociating member accounts (Security Hub API, AWS CLI)

To disassociate member accounts, you can use an API call or the AWS Command Line Interface.

To disassociate member accounts (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisassociateMembers` operation. You must provide the AWS account IDs for the member accounts to disassociate. To view a list of member accounts, use the `ListMembers` operation.
- **AWS CLI** – At the command line, run the `disassociate-members` command.

```
aws securityhub disassociate-members --account-ids <accountIds>
```

Example

```
aws securityhub disassociate-members --account-ids "123456789111" "123456789222"
```

Deleting member accounts

As an administrator account, you can delete member accounts that were added by invitation. Before you can delete an enabled account, you must disassociate it.

When you delete a member account, it is completely removed from the list. To restore the account's membership, you must add it and invite it as if it were a completely new member account.

Deleting member accounts (console)

From the Security Hub console, you can delete one or more accounts.

To delete member accounts

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. Under **Member accounts**, select the accounts to delete.
4. Choose **Actions**, and then choose **Delete accounts**.

Deleting member accounts (Security Hub API, AWS CLI)

To delete member accounts, you can use an API call or the AWS Command Line Interface.

To delete member accounts (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DeleteMembers` operation. You must provide the AWS account IDs of the member accounts to delete. To retrieve the list of member accounts, use the `ListMembers` operation.
- **AWS CLI** – At the command line, run the `delete-members` command.

```
aws securityhub delete-members --account-ids <memberAccountIDs>
```

Example

```
aws securityhub delete-members --account-ids "123456789111" "123456789222"
```

Disassociating from your administrator account

If your account was added as a member account by invitation, you can disassociate the member account from the administrator account. Once you disassociate a member account, Security Hub doesn't send findings from the account to the administrator account. Member accounts that are managed using Organizations cannot disassociate their accounts from the administrator account.

When you disassociate from your administrator account, your account remains in the administrator account's member list with a status of **Resigned**. However, the administrator account does not receive any findings for your account.

After you disassociate yourself from the administrator account, you can accept the invitation again.

Disassociating from an administrator account (console)

You can decline an invitation to be a member account. To do this, you update the **Accept** option for the administrator account.

To disassociate from your administrator account

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**, and then choose **Accounts**.
3. Under **Administrator account**, toggle **Accept** to the off position, and then choose **Update**.

Disassociating from an administrator account (Security Hub API, AWS CLI)

To disassociate your account from your administrator account, you can use an API call or the AWS Command Line Interface.

To disassociate from your administrator account (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisassociateFromAdministratorAccount` operation.
- **AWS CLI** – At the command line, run the `disassociate-from-administrator-account` command.

```
aws securityhub disassociate-from-administrator-account
```

Note

The Security Hub console continues to use `DisassociateFromMasterAccount`. It will eventually change to use `DisassociateFromAdministratorAccount`. Any IAM policies that specifically control access to this function must continue to use `DisassociateFromMasterAccount`. You should also add `DisassociateFromAdministratorAccount` to your policies to ensure that the correct permissions are in place after the console begins to use `DisassociateFromAdministratorAccount`.

Effect of account actions on Security Hub data

These account actions have the following effects on AWS Security Hub data.

Security Hub disabled

When you disable Security Hub for an account, it is disabled only for that account in the AWS Region that is selected when you disable it.

You must disable Security Hub separately in each Region where you enabled it.

No new findings are generated for the administrator account while Security Hub is disabled. Existing findings are deleted after 90 days.

Integrations with Amazon Macie, Amazon GuardDuty, and Amazon Inspector are removed.

Other Security Hub data and settings, including custom actions, insights, and subscriptions to third-party products are not removed.

Enabled security standards are disabled.

Member account disassociated from administrator account

When a member account is disassociated from the administrator account, the administrator account loses permission to view findings in the member account.

Security Hub continues to run in both accounts.

Custom settings or integrations that are defined for the administrator account are not applied to findings from the former member account. For example, after the accounts are disassociated, you might have a custom action in the administrator account used as the event pattern in an Amazon EventBridge rule. However, this custom action cannot be used in the member account.

Member account is removed from an organization

When a member account is removed from an organization, the Security Hub administrator account loses permission to view findings in the member account.

Security Hub continues to run in both accounts.

In the **Accounts** list for the Security Hub administrator account, the account has a status of **Disassociated**.

Account is suspended

When an account is suspended in AWS, the account loses permission to view their findings in Security Hub. No new findings are generated for that account. The administrator account for a suspended account can view the existing account findings.

For an organization account, the member account status can also change to **Account Suspended**. This happens if the account is suspended at the same time that the administrator account attempts to enable the account. The administrator account for an **Account Suspended** account cannot view findings for that account.

Otherwise, the suspended status does not affect the member account status.

After 90 days, the account is either terminated or reactivated. When the account is reactivated, its Security Hub permissions are restored. If the member account status is **Account Suspended**, the administrator account must enable the account manually.

Account is closed

When an AWS account is closed, Security Hub responds to the closure as follows.

Security Hub retains the findings for the account for 90 days from the effective date of the account closure. At the end of the 90 day period, Security Hub permanently deletes all findings for the account.

- To retain findings for more than 90 days, you can use a custom action with an EventBridge rule to store the findings in an Amazon S3 bucket. As long as Security Hub retains the findings, when you reopen the closed account, Security Hub restores the findings for the account.
- If the account is a Security Hub administrator account, it is removed as an administrator and all the member accounts are removed. If the account is a member account, it is disassociated and removed as a member from the Security Hub administrator account.

- For more information, see [Closing an account](#).

Important

For customers in the AWS GovCloud (US) Regions:

- Before closing your account, back up and then delete your policy data and other account resources. You will no longer have access to them after you close the account.

Cross-Region aggregation

With cross-Region aggregation, you can aggregate findings, insights, control compliance statuses, and security scores from multiple Regions to a single aggregation Region. You can then manage all of this data from the aggregation Region.

Note

Cross-Region aggregation is not supported in AWS GovCloud (US). In the China Regions, aggregation is only supported for findings and insights across the China Regions (specifically, you can aggregate findings and insights from China (Beijing) to China (Ningxia), or vice versa).

For example, suppose you set US East (N. Virginia) as an aggregation Region, and US West (Oregon) and US West (N. California) as your linked Regions. When you view the **Findings** page in US East (N. Virginia), you see the findings from all three Regions.

In the aggregation Region, the **Summary** page provides a view of your active findings across linked Regions. See [the section called "Viewing a cross-Region finding summary" \(p. 125\)](#). Other **Summary** page panels that analyze findings also display information from across the linked Regions.

Your security scores in the aggregation Region are calculated by comparing the number of passed controls to the number of enabled controls in all linked Regions. In addition, if a control is enabled in at least one linked Region, it is visible on the **Security standards** details pages of the aggregation Region. The compliance status of controls on the standards details pages reflects findings across linked Regions. If a security check associated with a control fails in one or more linked Regions, the compliance status of that control shows as **Failed** on the standards details pages of the aggregation Region. The number of security checks includes findings from all linked Regions.

The control enabled or disabled status is regional and must be modified in each Region. If a control is enabled in a linked Region but disabled in the aggregation Region, you can see the compliance status of that control from the aggregation Region, but you cannot enable or disable that control from the aggregation Region.

To view cross-Region security scores and compliance statuses, add the following permissions to your IAM policies:

- `ListSecurityControlDefinitions`
- `BatchGetStandardsControlAssociations`
- `BatchUpdateStandardsControlAssociations`

Note that these IAM permission names do not directly correspond to current Security Hub APIs .

Topics

- [How cross-Region aggregation works \(p. 112\)](#)
- [Viewing the current cross-Region aggregation configuration \(p. 113\)](#)
- [Enabling cross-Region aggregation \(p. 114\)](#)
- [Updating the cross-Region aggregation configuration \(p. 115\)](#)
- [Stopping cross-Region aggregation \(p. 116\)](#)

How cross-Region aggregation works

Cross-Region aggregation is configured by standalone accounts and by administrator accounts. Member accounts inherit the cross-Region aggregation configuration from their administrator account.

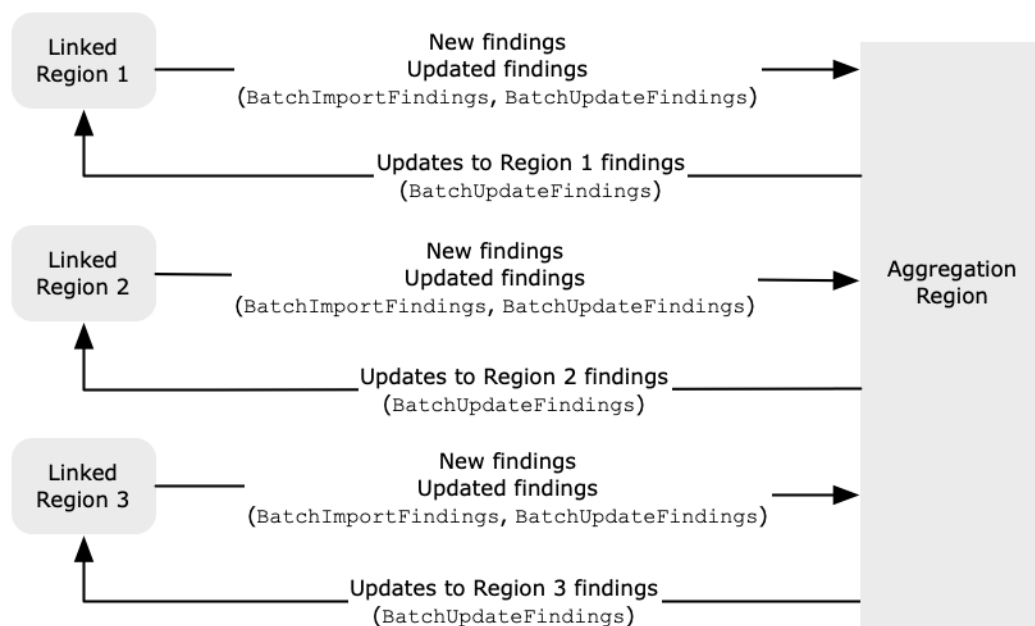
When a member account is disassociated from the administrator account, then cross-Region aggregation is stopped for the member account. This is true even if the account had enabled cross-Region aggregation before it became a member account.

Cross-Region aggregation is based on an aggregation Region and linked Regions.

Aggregating new data and replicating updates to data

When cross-Region aggregation is enabled, Security Hub aggregates new findings, insights, control compliance statuses, and security scores from the linked Regions to the aggregation Region.

Security Hub also replicates updates to this data between the linked Regions and the aggregation Region. Updates that occur in a linked Region are replicated to the aggregation Region. Updates that occur in the aggregation Region are replicated back to the original linked Region.



If there are conflicting updates in the aggregation Region and the linked Region, then the most recent update is used.

Cross-Region aggregation does not add to the cost of Security Hub. You are not charged when Security Hub replicates new data or updates.

Determining the accounts to aggregate data from

Security Hub only aggregates data from Regions where an account has Security Hub enabled. Security Hub is not automatically enabled for an account based on the cross-Region aggregation configuration.

When an administrator account configures cross-Region aggregation, Security Hub identifies the member accounts for that administrator account in the linked Regions.

In each linked Region, every member account for that administrator account inherits the cross-Region aggregation configuration. Security Hub aggregates their findings, insights, control statuses, and security scores to the aggregation Region.

If a member account from the aggregation Region is not a member account in a linked Region, then Security Hub does not aggregate data for that account from that Region.

If you plan to use cross-Region aggregation, and have multiple administrator accounts, then Security Hub recommends the following best practices:

- Each administrator account has the same member accounts across Regions.
- Each administrator account has different member accounts.
- Each administrator account uses a different aggregation Region.

Viewing the current cross-Region aggregation configuration

You can view the current cross-Region aggregation configuration from any Region. The configuration includes the aggregation Region, the linked Regions, and whether to automatically link new Regions.

Viewing the cross-Region aggregation configuration (console)

The **Regions** tab of the **Settings** page displays the current cross-Region aggregation configuration. You can view the configuration from any Region. Member accounts can also view the cross-Region configuration that the administrator account configured.

If cross-Region aggregation is not enabled, then the **Regions** tab displays the option to enable cross-Region aggregation. See [the section called "Enabling cross-Region aggregation" \(p. 114\)](#). Only administrator accounts and standalone accounts can enable cross-Region aggregation.

If cross-Region aggregation is enabled, then the **Regions** tab displays the following information:

- The aggregation Region
- Whether to automatically aggregate findings, insights, control statuses, and security scores from new Regions that Security Hub supports and that you opt into
- The list of linked Regions

Viewing the current cross-Region aggregation configuration (Security Hub API, AWS CLI)

You can use the Security Hub API or AWS CLI to view the current cross-Region aggregation configuration. You can view the cross-Region aggregation configuration from any Region.

To view the current cross-Region aggregation configuration (Security Hub API, AWS CLI)

- **Security Hub API:** Use the [GetFindingAggregator](#) API. When you make the request, you must provide the finding aggregator ARN. To obtain the finding aggregator ARN, use [ListFindingAggregators](#).

- **AWS CLI:** At the command line, run the `get-finding-aggregator` command. To obtain the finding aggregator ARN, use `list-finding-aggregators`.

```
aws securityhub get-finding-aggregator --finding-aggregator-arn <finding aggregator ARN>
```

Enabling cross-Region aggregation

You must enable cross-Region aggregation from the Region that will be the aggregation Region.

You cannot use a Region that is disabled by default as your aggregation Region. For a list of Regions that are disabled by default, see [Enabling a Region](#) in the *AWS General Reference*.

Enabling cross-Region aggregation (console)

When you enable cross-Region aggregation, you choose your linked Regions. You also choose whether to automatically link new Regions when Security Hub begins to support them and you have opted into them.

To enable cross-Region aggregation

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Change to the Region that you want to use as the aggregation Region.
3. In the Security Hub navigation menu, choose **Settings**, then choose **Regions**.
4. Under **Finding aggregation**, choose **Configure finding aggregation**.

By default, the aggregation Region is set to **No aggregation Region**.

5. Under **Aggregation Region**, choose the radio button to designate the current Region as the aggregation Region.
6. Under **Linked Regions**, select the Regions to aggregate data from.
7. To automatically aggregate data from new Regions in the partition as Security Hub supports them and you opt into them, select **Link future Regions**.
8. Choose **Save**.

Enabling cross-Region aggregation (Security Hub API, AWS CLI)

You can use the Security Hub API to enable cross-Region aggregation.

To enable cross-Region aggregation from the Security Hub API, you create a finding aggregator. You must create the finding aggregator from the Region that you want to use as the aggregation Region.

To create the finding aggregator (Security Hub API, AWS CLI)

- **Security Hub API:** From the Region that you want to use as the aggregation Region, use the `CreateFindingAggregator` operation. For `RegionLinkingMode`, you choose from the following options:
 - `ALL_REGIONS` – Security Hub aggregates data from all Regions. Security Hub also aggregates data from new Regions as they are supported and you opt into them.
 - `ALL_REGIONS_EXCEPT_SPECIFIED` – Security Hub aggregates data from all Regions except for Regions that you want to exclude. Security Hub also aggregates data from new Regions as they

are supported and you opt into them. Use `Regions` to provide the list of Regions to exclude from aggregation.

- `SPECIFIED_REGIONS` – Security Hub aggregates data from a selected list of Regions. Security Hub does not aggregate data automatically from new Regions. Use `Regions` to provide the list of Regions to aggregate from.
- **AWS CLI:** At the command line, run the `create-finding-aggregator` command.

```
aws securityhub create-finding-aggregator --region <aggregation Region> --region-linking-mode ALL_REGIONS | ALL_REGIONS_EXCEPT_SPECIFIED | SPECIFIED_REGIONS --regions <Region list>
```

In the following example, cross-Region aggregation is configured for selected Regions. The aggregation Region is US East (N. Virginia). The linked Regions are US West (N. California) and US West (Oregon).

```
aws securityhub create-finding-aggregator --region us-east-1 --region-linking-mode SPECIFIED_REGIONS --regions us-west-1,us-west-2
```

Updating the cross-Region aggregation configuration

You can update the cross-Region aggregation configuration to change the linked Regions for the current aggregation Region. You can also change whether to automatically aggregate findings, insights, control statuses, and security scores from new Regions.

When you stop aggregating data from a linked Region, Security Hub does not remove any existing aggregated data from the aggregation Region.

You cannot use the update process to change the aggregation Region. To change the aggregation Region, you must do the following:

1. Stop cross-Region aggregation. See [the section called “Stopping cross-Region aggregation” \(p. 116\)](#).
2. Change to the Region that you want to be the new aggregation Region.
3. Enable cross-Region aggregation. See [the section called “Enabling cross-Region aggregation” \(p. 114\)](#).

Updating the cross-Region aggregation configuration (console)

You must update the cross-Region aggregation configuration from the current aggregation Region.

In Regions other than the aggregation Region, the **Finding aggregation** panel displays a message that you must edit the configuration in the aggregation Region. Choose this message to display a link to navigate to the aggregation Region.

To change the linked Regions for the current aggregation Region

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Change to the current aggregation Region.
3. In the Security Hub navigation menu, choose **Settings**, then choose **Regions**.

4. Under **Finding aggregation**, choose **Edit**.
5. Under **Linked Regions**, update the selected linked Regions.
6. If needed, change whether **Link future Regions** is selected. This setting determines whether Security Hub automatically links new Regions as it adds support for them and you opt into them.
7. Choose **Save**.

Updating the cross-Region aggregation configuration (Security Hub API, AWS CLI)

You can use the Security Hub API or AWS CLI to update the cross-Region aggregation configuration. You must update cross-Region aggregation from the current aggregation Region.

You can change the Region linking mode. If the linking mode is `ALL_REGIONS_EXCEPT_SPECIFIED` or `SPECIFIED_REGIONS`, you can change the list of excluded or included Regions.

When you change the list of excluded or included Regions, you must provide the full list with the updates. For example, suppose you currently aggregate findings from US East (Ohio), and want to also aggregate findings from US West (Oregon). When you call `UpdateFindingAggregator`, you provide a `Regions` list that contains both US East (Ohio) and US West (Oregon).

To update cross-Region aggregation (Security Hub API, AWS CLI)

- **Security Hub API:** Use the `UpdateFindingAggregator` API operation. To identify the finding aggregator, you must provide the finding aggregator ARN. To obtain the finding aggregator ARN, use [ListFindingAggregators](#).

You provide the Region linking mode and the updated list of excluded or included Regions.

- **AWS CLI:** At the command line, run the `update-finding-aggregator` command.

```
aws securityhub update-finding-aggregator --region <aggregation Region> --finding-aggregator-arn <finding aggregator ARN> --region-linking-mode ALL_REGIONS | ALL_REGIONS_EXCEPT_SPECIFIED | SPECIFIED_REGIONS --regions <Region list>
```

In the following example, the cross-Region aggregation configuration is changed to aggregation for selected Regions. The command is run from the current aggregation Region, which is US East (N. Virginia). The linked Regions are US West (N. California) and US West (Oregon).

```
aws securityhub update-finding-aggregator --region us-east-1 --finding-aggregator-arn arn:aws:securityhub:us-east-1:222222222222:finding-aggregator/123e4567-e89b-12d3-a456-426652340000 --region-linking-mode SPECIFIED_REGIONS --regions us-west-1,us-west-2
```

Stopping cross-Region aggregation

Stop cross-Region aggregation if you no longer want to aggregate data or if you want to change the aggregation Region.

When you stop cross-Region aggregation, Security Hub stops aggregating data. It does not remove any existing aggregated data from the aggregation Region.

Stopping cross-Region aggregation (console)

You must stop cross-Region aggregation from the current aggregation Region.

In Regions other than the aggregation Region, the **Finding aggregation** panel displays a message that you must edit the configuration in the aggregation Region. Choose this message to display a link to switch to the aggregation Region.

To stop cross-Region aggregation

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Change to the current aggregation Region.
3. In the Security Hub navigation menu, choose **Settings**, then choose **Regions**.
4. Under **Finding aggregation**, choose **Edit**.
5. Under **Aggregation Region**, choose **No aggregation Region**.
6. Choose **Save**.
7. On the confirmation dialog, in the confirmation field, type **Confirm**.
8. Choose **Confirm**.

Stopping cross-Region aggregation (Security Hub API, AWS CLI)

You can use the Security Hub API to stop cross-Region aggregation. You must stop cross-Region aggregation from the aggregation Region.

To stop cross-Region aggregation (Security Hub API, AWS CLI)

- **Security Hub API:** Use the `DeleteFindingAggregator` operation. To identify the finding aggregator to delete, you provide the finding aggregator ARN. To obtain the finding aggregator ARN, use `ListFindingAggregators`.
- **AWS CLI:** At the command line, run the `delete-finding-aggregator` command.

```
aws securityhub delete-finding-aggregator <finding aggregator ARN> --region <aggregation Region>
```

Findings in AWS Security Hub

AWS Security Hub eliminates the complexity of addressing large volumes of findings from multiple providers. It reduces the effort required to manage and improve the security of all of your AWS accounts, resources, and workloads.

Security Hub receives findings from the following sources.

- Integrations with AWS security services that you enable. See [the section called “AWS service integrations” \(p. 293\)](#).
- Integrations with third-party products that you enable. See [the section called “Third-party product integrations” \(p. 304\)](#).
- Custom integrations that you configure. See [the section called “Using custom product integrations” \(p. 321\)](#).
- Security Hub checks against enabled controls. See [the section called “Generating and updating control findings” \(p. 325\)](#).

Security Hub consumes findings using a standard findings format called the AWS Security Finding Format. For more information about the finding format, see [the section called “Finding format” \(p. 132\)](#).

Security Hub correlates the findings across integrated products to prioritize the most important ones.

Finding providers can update findings to reflect additional instances of the finding. You can update findings to provide details about your investigation and its results.

Security Hub also allows you to aggregate findings across Regions, so that you can view all of your findings from one place. See [Cross-Region aggregation \(p. 111\)](#).

Topics

- [Creating and updating findings in AWS Security Hub \(p. 118\)](#)
- [Viewing a cross-Region summary of findings by severity \(p. 125\)](#)
- [Viewing finding lists and details in AWS Security Hub \(p. 126\)](#)
- [Taking action on findings in AWS Security Hub \(p. 130\)](#)
- [AWS Security Finding Format \(ASFF\) \(p. 132\)](#)

Creating and updating findings in AWS Security Hub

In AWS Security Hub, a finding can originate from one of the following types of finding providers.

- An enabled integration with another AWS service
- An enabled integration with a third-party provider
- An enabled control in Security Hub

After a finding is created, it can be updated by the finding provider or by the customer.

- The finding provider uses the `BatchImportFindings` API operation to update the general information about a finding. Finding providers can only update findings that they created.
- The customer uses the `BatchUpdateFindings` API operation to reflect the status of the investigation into a finding. `BatchUpdateFindings` can also be used by a ticketing, incident management, orchestration, remediation, or SIEM tool on behalf of the customer.

From the Security Hub console, customers can manage the workflow status of findings and send findings to custom actions. See [the section called "Taking action on findings" \(p. 130\)](#).

Security Hub also automatically updates and deletes findings.

All findings are automatically deleted if they were not updated in the past 90 days.

If you enable cross-Region aggregation, then Security Hub automatically aggregates new findings from the linked Regions to the aggregation Region. Security Hub also replicates updates to findings. Updates that occur in the linked Regions are replicated to the aggregation Region. Updates that occur in the aggregation Region are replicated to the original linked Region. For more information about cross-Region aggregation, see [Cross-Region aggregation \(p. 111\)](#).

Topics

- [Using BatchImportFindings to create and update findings \(p. 119\)](#)
- [Using BatchUpdateFindings to update a finding \(p. 122\)](#)

Using BatchImportFindings to create and update findings

Finding providers use the `BatchImportFindings` API operation to create new findings and to update information about the findings they created. They cannot update findings that they did not create.

Customers, SIEMs, ticketing tools, and SOAR tools use `BatchUpdateFindings` to make updates related to their processing of findings from finding providers. See [the section called "Using BatchUpdateFindings" \(p. 122\)](#).

Whenever AWS Security Hub receives a `BatchImportFindings` request to either create or update a finding, it automatically generates a **Security Hub Findings - Imported** event in Amazon EventBridge. See [Automated response and remediation \(p. 604\)](#).

Requirements for accounts and batch size

`BatchImportFindings` must be called by one of the following:

- The account that is associated with the findings. The identifier of the associated account is the value of the `AwsAccountId` attribute for the finding.
- An account that is allow-listed for an official Security Hub partner integration.

Security Hub can only accept finding updates for accounts that have Security Hub enabled. The finding provider also must be enabled. If Security Hub is disabled, or the finding provider integration is not enabled, then the findings are returned in the `FailedFindings` list, with an `InvalidAccess` error.

`BatchImportFindings` accepts up to 100 findings per batch, up to 240 KB per finding, and up to 6 MB per batch. The throttle rate limit is 10 TPS per account per Region, with a burst of 30 TPS.

Determining whether to create or update a finding

To determine whether to create or update a finding, Security Hub checks the `ID` field. If the value of `ID` does not match an existing finding, then a new finding is created.

If `ID` does match an existing finding, then Security Hub checks the `UpdatedAt` field for the update.

- If `UpdatedAt` on the update matches or occurs before `UpdatedAt` on the existing finding, then the update is ignored.
- If `UpdatedAt` on the update occurs after `UpdatedAt` on the existing finding, then the existing finding is updated.

Restricted attributes for BatchImportFindings

For an existing finding, finding providers cannot use `BatchImportFindings` to update the following attributes and objects. These attributes can only be updated using `BatchUpdateFindings`.

- `Note`
- `UserDefinedFields`
- `VerificationState`
- `Workflow`

Security Hub ignores any content in `BatchImportFindings` for those attributes and objects. Customers, or other providers acting on their behalf, use `BatchUpdateFindings` to update them.

Using FindingProviderFields

Finding providers also should not use `BatchImportFindings` to update the following attributes.

- `Confidence`
- `Criticality`
- `RelatedFindings`
- `Severity`
- `Types`

Instead, finding providers use the `FindingProviderFields` (p. 189) object to provide values for these attributes.

Example

```
"FindingProviderFields": {
  "Confidence": 42,
  "Criticality": 99,
  "RelatedFindings": [
    {
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
      "Id": "123e4567-e89b-12d3-a456-426655440000"
    }
  ],
  "Severity": {
    "Label": "MEDIUM",
    "Original": "MEDIUM"
  },
}
```



```
"Types": [ "Software and Configuration Checks/Vulnerabilities/CVE" ]  
}
```

For `BatchImportFindings` requests, Security Hub handles values in the top-level attributes and in `FindingProviderFields` (p. 189) as follows.

(Preferred) `BatchImportFindings` provides a value for an attribute in `FindingProviderFields` (p. 189), but does not provide a value for the corresponding top-level attribute.

For example, `BatchImportFindings` provides `FindingProviderFields.Confidence`, but does not provide `Confidence`. This is the preferred option for `BatchImportFindings` requests.

Security Hub updates the value of the attribute in `FindingProviderFields` (p. 189).

It replicates the value to the top-level attribute only if the attribute was not already updated by `BatchUpdateFindings`.

`BatchImportFindings` provides a value for a top-level attribute, but does not provide a value for the corresponding attribute in `FindingProviderFields` (p. 189).

For example, `BatchImportFindings` provides `Confidence`, but does not provide `FindingProviderFields.Confidence`.

Security Hub uses the value to update the attribute in `FindingProviderFields` (p. 189). It overwrites any existing value.

Security Hub updates the top-level attribute only if the attribute was not already updated by `BatchUpdateFindings`.

`BatchImportFindings` provides a value for both a top-level attribute and the corresponding attribute in `FindingProviderFields` (p. 189).

For example, `BatchImportFindings` provides both `Confidence` and `FindingProviderFields.Confidence`.

For a new finding, Security Hub uses the value in `FindingProviderFields` (p. 189) to populate both the top-level attribute and the corresponding attribute in `FindingProviderFields` (p. 189). It does not use the provided top-level attribute value.

For an existing finding, Security Hub uses both values. However, it updates the top-level attribute value only if the attribute was not already updated by `BatchUpdateFindings`.

Using the batch-import-findings command from the AWS CLI

In the AWS Command Line Interface, you use the `batch-import-findings` command to create or update findings.

You provide each finding as a JSON object.

Example

```
aws securityhub batch-import-findings --findings '  
  [{  
    "AwsAccountId": "123456789012",  
    "CreatedAt": "2019-08-07T17:05:54.832Z",  
    "Description": "Vulnerability in a CloudTrail trail",  
    "FindingProviderFields": {
```

```
    "Severity": {
      "Label": "INFORMATIONAL",
      "Original": "0"
    },
    "Types": [
      "Software and Configuration Checks/Vulnerabilities/CVE"
    ]
  },
  "GeneratorId": "TestGeneratorId",
  "Id": "Id1",
  "ProductArn": "arn:aws:securityhub:us-west-1:123456789012:product/123456789012/default",
  "Resources": [
    {
      "Id": "arn:aws:cloudtrail:us-west-1:123456789012:trail/TrailName",
      "Partition": "aws",
      "Region": "us-west-1",
      "Type": "AwsCloudTrailTrail"
    }
  ],
  "SchemaVersion": "2018-10-08",
  "Title": "CloudTrail trail vulnerability",
  "UpdatedAt": "2020-06-02T16:05:54.832Z"
}]'
```

Using BatchUpdateFindings to update a finding

[BatchUpdateFindings](#) is used to update information related to a customer's processing of findings from finding providers. It can be used by a customer or by a SIEM, ticketing, incident management, or SOAR tool that works on behalf of a customer. [BatchUpdateFindings](#) cannot be used to create new findings. It can be used to update up to 100 findings at a time.

Whenever Security Hub receives a [BatchUpdateFindings](#) request to update a finding, it automatically generates a **Security Hub Findings - Imported** event in Amazon EventBridge. See [Automated response and remediation](#) (p. 604).

[BatchUpdateFindings](#) does not change the `UpdatedAt` field for the finding. `UpdatedAt` only reflects the most recent update from the finding provider.

Available fields for BatchUpdateFindings

Administrator accounts can use [BatchUpdateFindings](#) to update findings for their account or for their member accounts. Member accounts can use [BatchUpdateFindings](#) to update findings for their account.

Customers can only use [BatchUpdateFindings](#) to update the following fields and objects.

- Confidence
- Criticality
- Note
- RelatedFindings
- Severity
- Types
- UserDefinedFields
- VerificationState
- Workflow

By default, administrator and member accounts have access to all of the above fields and field values. Security Hub also provides context keys to allow you to restrict access to fields and field values.

For example, you might only allow member accounts to set `Workflow.Status` to `RESOLVED`. Or you might not want to allow member accounts to change `Severity.Label`.

Configuring access to BatchUpdateFindings

You can configure IAM policies to restrict access to using `BatchUpdateFindings` to update fields and field values.

In a statement to restrict access to `BatchUpdateFindings`, use the following values.

- Action is `securityhub:BatchUpdateFindings`
- Effect is `Deny`
- For Condition, you can deny a `BatchUpdateFindings` request based on the following:
 - The finding includes a specific field.
 - The finding includes a specific field value.

Condition keys

These are the condition keys for restricting access to `BatchUpdateFindings`.

ASFF field

The condition key for an ASFF field is as follows.

```
securityhub:ASFFSyntaxPath/<fieldName>
```

Replace `<fieldName>` with the ASFF field.

For example, to restrict access to the `Workflow.Status` field, use `securityhub:ASFFSyntaxPath/Workflow.Status`.

Disallowing all updates to a field

To prevent a user from making any update to a specific field, use a condition like this:

```
"Condition": {
  "Null": {
    "securityhub:ASFFSyntaxPath/<fieldName>": "false"
  }
}
```

For example, the following statement indicates that `BatchUpdateFindings` cannot be used to update the workflow status.

```
{
  "Sid": "VisualEditor0",
  "Effect": "Deny",
  "Action": "securityhub:BatchUpdateFindings",
  "Resource": "*",
  "Condition": {
    "Null": {
      "securityhub:ASFFSyntaxPath/Workflow.Status": "false"
    }
  }
}
```

```
    }  
  }  
}
```

Disallowing specific field values

To prevent a user from setting a field to a specific value, use a condition like this:

```
"Condition": {  
  "StringEquals": {  
    "securityhub:ASFFSyntaxPath/<fieldName>": "<fieldValue>"  
  }  
}
```

For example, the following statement indicates that `BatchUpdateFindings` cannot be used to set `Workflow.Status` to `SUPPRESSED`.

```
{  
  "Sid": "VisualEditor0",  
  "Effect": "Deny",  
  "Action": "securityhub:BatchUpdateFindings",  
  "Resource": "*",  
  "Condition": {  
    "StringEquals": {  
      "securityhub:ASFFSyntaxPath/Workflow.Status": "SUPPRESSED"  
    }  
  }  
}
```

You can also provide a list of values that are not permitted.

```
"Condition": {  
  "ForAnyValue:StringEquals": {  
    "securityhub:ASFFSyntaxPath/<fieldName>": [ "<fieldValue1>",  
    "<fieldValue2>", "<fieldValueN>" ]  
  }  
}
```

For example, the following statement indicates that `BatchUpdateFindings` cannot be used to set `Workflow.Status` to either `RESOLVED` or `SUPPRESSED`.

```
{  
  "Sid": "VisualEditor0",  
  "Effect": "Deny",  
  "Action": "securityhub:BatchUpdateFindings",  
  "Resource": "*",  
  "Condition": {  
    "ForAnyValue:StringEquals": {  
      "securityhub:ASFFSyntaxPath/Workflow.Status": [  
        "RESOLVED",  
        "NOTIFIED"  
      ]  
    }  
  }  
}
```

Using the batch-update-findings command from the AWS CLI

In the AWS Command Line Interface, you use the `batch-update-findings` command to update the findings.

For each finding to update, you provide both the finding ID and the ARN of the product that generated the finding.

```
--finding-identifiers ID="<findingID1>",ProductArn="<productARN>"  
ID="<findingID2>",ProductArn="<productARN2>"
```

When you provide the attributes to update, you can either use a JSON format or a shortcut format.

Here is an example of an update to the Note object that uses the JSON format:

```
--note '{"Text": "Known issue that is not a risk.", "UpdatedBy": "user1"}'
```

Here is the same update that uses the shortcut format:

```
--note Text="Known issue that is not a risk.",UpdatedBy="user1"
```

The AWS CLI Command Reference provides the JSON and shortcut syntax for each field.

The following [batch-update-findings](#) example updates two findings to add a note, change the severity label, and resolve them.

```
aws securityhub batch-update-findings --finding-identifiers Id="arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-west-2:product/aws/securityhub"  
Id="arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/  
finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-  
west-1:product/aws/securityhub" --note '{"Text": "Known issue that is not a risk.",  
"UpdatedBy": "user1"}' --severity '{"Label": "LOW"}' --workflow '{"Status": "RESOLVED"}'
```

This is the same example, but uses the shortcuts instead of JSON.

```
aws securityhub batch-update-findings --finding-identifiers Id="arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-west-1:product/aws/securityhub"  
Id="arn:aws:securityhub:us-west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/  
finding/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",ProductArn="arn:aws:securityhub:us-  
west-1:product/aws/securityhub" --note Text="Known issue that is not a  
risk.",UpdatedBy="user1" --severity Label="LOW" --workflow Status="RESOLVED"
```

Viewing a cross-Region summary of findings by severity

On the **Summary** page, **Findings by Region** summarizes the number of active findings for each severity across Regions. The counts only include findings that have a workflow status of **NEW** or **NOTIFIED**.

Before you enable cross-Region aggregation, **Findings by Region** only summarizes the findings for the current Region. It provides a link to configure cross-Region aggregation.

After you enable cross-Region aggregation, **Findings by Region** only displays in the aggregation Region. It does not display in linked Regions. The panel summarizes the findings for all of the linked Regions.

By default, the list only displays linked Regions that have matching findings. To display all of the linked Regions, including linked Regions that don't have matching findings, choose **All linked Regions**. To only display linked Regions that have matching findings, choose **Linked Regions with findings**.

When you choose a severity value for a Region, the **Findings** list is displayed. The list is filtered by the selected Region and severity. The list also is filtered to display active findings that have a workflow status of `NEW` or `NOTIFIED`.

Viewing finding lists and details in AWS Security Hub

In the AWS Security Hub navigation pane, **Findings** displays a list of findings from all of the enabled product integrations and controls.

From **Integrations**, you can display a list of findings generated by an enabled integration. See [the section called “Viewing the findings from an integration” \(p. 293\)](#).

From **Insights**, you can display a list of findings for a matching insight result. See [the section called “Viewing insight results and findings” \(p. 277\)](#).

From **Security standards**, you can display a list of findings generated from a selected control. See [the section called “Viewing and taking action on control findings” \(p. 343\)](#).

You can also use the `GetFindings` API operation to retrieve a filtered list of findings.

If you enable finding aggregation, you can view findings from across Regions. In the aggregation Region, the **Findings** and **Insights** pages contain findings from the aggregation Region and the linked Regions. In other Regions, these pages only contain findings from that Region. For information on how to configure finding aggregation, see [Cross-Region aggregation \(p. 111\)](#).

Topics

- [Filtering and grouping findings \(console\) \(p. 126\)](#)
- [Viewing finding details \(console\) \(p. 128\)](#)
- [Retrieving finding details \(Security Hub API, AWS CLI\) \(p. 129\)](#)

Filtering and grouping findings (console)

When you display a list of findings from the **Findings** page, the **Integrations** page, or the **Insights** page, the list is always filtered based on the record state and workflow status. This is in addition to the filters for an insight or integration.

The record state indicates whether the finding is active or archived. A finding can be archived by the finding provider. AWS Security Hub also automatically archives findings for controls if the associated resource is deleted. By default, a finding list only shows active findings.

The workflow status indicates the status of the investigation into the finding. The workflow status can only be updated by the Security Hub customer or a system that is operating on the customer's behalf. By default, a finding list only shows findings with a workflow status of `NEW` or `NOTIFIED`. The default finding list for a control also includes `RESOLVED` findings.

If you enabled finding aggregation, then on the **Findings** and **Insights** pages, you can filter the findings by Region.

For information on working with the finding list for a control, see [the section called “Filtering and sorting findings” \(p. 344\)](#).

Adding filters

To change the scope of the list, you can add filters to it.

You can filter by up to 10 attributes. For each attribute, you can provide up to 20 filter values.

When filtering the finding list, Security Hub applies AND logic to the set of filters. In other words, a finding only matches if it matches all of the provided filters. For example, if you add `GuardDuty` as a filter for product name, and `AwsS3Bucket` as a filter for resource type, then matching findings must match both of these criteria.

However, Security Hub applies OR logic to filters that use the same attribute but different values. For example, you add both `GuardDuty` and `Amazon Inspector` as filter values for product name. In that case, a finding matches if it was generated by either `GuardDuty` or `Amazon Inspector`.

To add a filter to the finding list

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. To display a finding list, do one of the following:
 - In the Security Hub navigation pane, choose **Findings**.
 - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
 - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
3. Choose the **Add filters** box.
4. In the menu, under **Filters**, choose a filter.

Note that when you filter by **Company name** or **Product name**, Security Hub uses the top-level `CompanyName` and `ProductName` fields. The API uses the values that are in `ProductFields`.

5. Choose the filter match type.

For a string filter, you can choose from the following comparison options:

- **is** – Find a value that exactly matches the filter value.
- **starts with** – Find a value that starts with the filter value.
- **is not** – Find a value that does not match the filter value.
- **does not start with** – Find a value that does not start with the filter value.

For a numeric filter, you can choose whether to provide a single number (**Simple**) or a range of numbers (**Range**).

For a date or time filter, you can choose whether to provide a length of time from the current date and time (**Rolling window**) or a specific date range (**Fixed range**).

Adding multiple filters has the following interactions:

- **is** and **starts with** filters are joined by OR. A value matches if it contains any of the filter values. For example, if you specify **Severity label is CRITICAL** and **Severity label is HIGH**, the results include both critical and high severity findings.
- **is not** and **does not start with** filters are joined by AND. A value matches only if it does not contain any of those filter values. For example, if you specify **Severity label is not LOW** and **Severity label is not MEDIUM**, the results do not include low or medium severity findings.

If you have an **is** filter on a field, you cannot have an **is not** or a **does not start with** filter on the same field.

6. Specify the filter value.

Note that for string filters, the filter value is case sensitive.

For example, for findings from Security Hub, **Product name** is Security Hub. If you use the **EQUALS** operator to see findings from Security Hub, you must enter **Security Hub** as the filter value. If you enter **security hub**, no findings are displayed.

Similarly, if you use the **PREFIX** operator, and enter **Sec**, Security Hub findings are displayed. If you enter **sec**, no Security Hub findings are displayed.

7. Choose **Apply**.

Grouping findings

In addition to changing the filters, you can group the findings based on the values of a selected attribute.

When you group the findings, the list of findings is replaced with a list of values for the selected attribute in the matching findings. For each value, the list displays the number of findings that match the other filter criteria.

For example, if you group the findings by AWS account ID, you see a list of account identifiers, with the number of matching findings for each account.

Note that Security Hub can only display 100 values. If there are more than 100 grouping values, you only see the first 100.

When you choose an attribute value, the list of matching findings for that value is displayed.

To group the findings in a findings list

1. On the finding list, choose the **Add filters** box.
2. In the menu, under **Grouping**, choose **Group by**.
3. In the list, choose the attribute to use for the grouping.
4. Choose **Apply**.

Changing a filter value or grouping attribute

For an existing filter, you can change the filter value. You can also change the grouping attribute.

For example, you can change the **Record state** filter to look for **ARCHIVED** findings instead of **ACTIVE** findings.

To edit a filter or grouping attribute

1. On a filtered finding list, choose the filter or grouping attribute.
2. For **Group by**, choose the new attribute, then choose **Apply**.
3. For a filter, choose the new value, and then choose **Apply**.

Deleting a filter or grouping attribute

To delete a filter or grouping attribute, choose the **x** icon.

The list is updated automatically to reflect the change. When you remove the grouping attribute, the list changes from the list of field values back to a list of findings.

Viewing finding details (console)

From a finding list, you can display a details pane for a finding.

To view the findings detail pane

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. To display a finding list, do one of the following:
 - In the Security Hub navigation pane, choose **Findings**.
 - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
 - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
3. Choose the finding title.

At the top of the finding details pane contains overview information about the finding, including the account, severity, dates, and status. If the account is an organization member account, then the information includes the account name. For accounts that are invited manually, the information only includes the account ID.

Types and Related Findings contains information about the finding type.

Resources contains information about the affected resource.

Remediation displays for control findings. It provides a link to the instructions for remediating the issue that triggered the finding.

Finding Provider Fields displays the values from the finding provider for confidence, criticality, related findings, severity, and finding type.

From the finding details pane, you can view more details and add field values to the filter.

- To display the complete JSON for the finding, choose the finding ID. From **Finding JSON**, you can download the finding JSON to a file.
- To add a field value to the finding list filter, choose the search icon next to the field.
- For findings that are based on AWS Config rules, to display a list of the applicable rules, choose **Rules**.

Retrieving finding details (Security Hub API, AWS CLI)

To retrieve details for selected findings programmatically, you can use an API call or the AWS Command Line Interface.

Note that when you filter by `CompanyName` or `ProductName`, Security Hub uses the values that are in `ProductFields`. It does not use the top-level `CompanyName` and `ProductName` fields.

To retrieve a list of findings (Security Hub API, AWS CLI)

- Security Hub API – Use the `GetFindings` API operation.
- AWS CLI – At the command line, run the `get-findings` command.

```
get-findings --filters <filter criteria JSON> --sort-criteria <sort criteria> --page-size <findings per page> --max-items <maximum number of results>
```

Example

```
aws securityhub get-findings --filters '{"GeneratorId":[{"Value": "aws-foundational", "Comparison": "PREFIX"}], "WorkflowStatus": [{"Value": "NEW", "Comparison": "EQUALS"}], "Confidence": [{"Gte": 85}]}' --sort-criteria '{"Field": "LastObservedAt", "SortOrder": "desc"}' --page-size 5 --max-items 100
```

Taking action on findings in AWS Security Hub

AWS Security Hub allows you to track the current status of your investigation into a finding.

You can also send findings to custom actions for processing.

Topics

- [Setting the workflow status for findings \(p. 130\)](#)
- [Sending findings to a custom action \(p. 131\)](#)

Setting the workflow status for findings

For findings, the workflow status tracks the progress of your investigation into a finding. The workflow status is specific to an individual finding. It does not affect the generation of new findings. For example, setting the workflow status to `SUPPRESSED` or `RESOLVED` does not prevent a new finding for the same issue.

The workflow status has the following values:

NEW

The initial state of a finding before you review it.

Security Hub also resets the workflow status from either `NOTIFIED` or `RESOLVED` to `NEW` in the following cases:

- `RecordState` changes from `ARCHIVED` to `ACTIVE`.
- `Compliance.Status` changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`.

These changes imply that additional investigation is required.

NOTIFIED

Indicates that you notified the resource owner about the security issue. You can use this status when you are not the resource owner, and you need intervention from the resource owner in order to resolve a security issue.

If one of the following occurs, the workflow status is changed automatically from `NOTIFIED` to `NEW`:

- `RecordState` changes from `ARCHIVED` to `ACTIVE`.
- `Compliance.Status` changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`.

SUPPRESSED

Indicates that you reviewed the finding and do not believe that any action is needed.

The workflow status of a `SUPPRESSED` finding does not change if `RecordState` changes from `ARCHIVED` to `ACTIVE`.

RESOLVED

The finding was reviewed and remediated and is now considered resolved.

The finding remains `RESOLVED` unless one of the following occurs:

- `RecordState` changes from `ARCHIVED` to `ACTIVE`.
- `Compliance.Status` changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`.

In those cases, the workflow status is automatically reset to `NEW`.

For findings from controls, if `Compliance.Status` is `PASSED`, then Security Hub automatically sets the workflow status to `RESOLVED`.

Setting the workflow status (console)

To set the workflow status from a finding details pane, from **Workflow status**, choose the status.

You can also set the workflow status for multiple selected findings in a finding list.

To set the workflow status for multiple findings (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. To display a finding list, do one of the following:
 - In the Security Hub navigation pane, choose **Findings**.
 - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
 - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
 - In the Security Hub navigation pane, choose **Security standards**. Choose **View results** to display a list of controls. Then choose the control name.
3. In the finding list, select the check box for each finding that you want to update.
4. At the top of the list, for **Workflow status**, choose the status.

Setting the workflow status (Security Hub API, AWS CLI)

To set the workflow status, you can use an API call or the AWS Command Line Interface.

To set the workflow status of a finding (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `BatchUpdateFindings` operation. To identify the finding to update, you must provide both the finding ID and the ARN of the product that generated the finding.
- **AWS CLI** – At the command line, run the `batch-update-findings` command.

```
batch-update-findings --finding-identifiers Id="<findingID>",ProductArn="<productARN>" --  
workflow Status="<workflowStatus>"
```

Example

```
aws securityhub batch-update-findings --finding-identifiers Id="arn:aws:securityhub:us-  
west-1:123456789012:subscription/pci-dss/v/3.2.1/PCI.Lambda.2/finding/a1b2c3d4-5678-90ab-  
cdef-EXAMPLE11111",ProductArn="arn:aws:securityhub:us-west-1:product/aws/securityhub" --  
workflow Status="RESOLVED"
```

Sending findings to a custom action

You can create AWS Security Hub custom actions to automate Security Hub with Amazon EventBridge. For custom actions, the event type is **Security Hub Findings - Custom Action**.

For more information and detailed steps on creating custom actions, see [Automated response and remediation \(p. 604\)](#).

After you set up a custom action, you can send findings to it.

To send findings to a custom action

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. To display a finding list, do one of the following:
 - In the Security Hub navigation pane, choose **Findings**.
 - In the Security Hub navigation pane, choose **Insights**. Choose an insight. Then on the results list, choose an insight result.
 - In the Security Hub navigation pane, choose **Integrations**. Choose **See findings** for an integration.
 - In the Security Hub navigation pane, choose **Security standards**. Choose **View results** to display a list of controls. Then choose the control name.
3. In the finding list, select the check box for each finding to send to the custom action.

You can send up to 20 findings at a time.
4. For **Actions**, choose the custom action.

AWS Security Finding Format (ASFF)

AWS Security Hub consumes, aggregates, organizes, and prioritizes findings from AWS security services and from the third-party product integrations. Security Hub processes these findings using a standard findings format called the AWS Security Finding Format (ASFF), which eliminates the need for time-consuming data conversion efforts. Then it correlates ingested findings across products to prioritize the most important ones.

Topics

- [ASFF syntax \(p. 132\)](#)
- [ASFF examples \(p. 176\)](#)
- [Types taxonomy for ASFF \(p. 274\)](#)

ASFF syntax

The following is a complete outline of the JSON for a finding in the AWS Security Finding Format (ASFF).

Choose a linked object name to view an example finding for that object.

To view descriptions of the required ASFF attributes, see [the section called “Required attributes” \(p. 176\)](#).

To view descriptions of the other top-level ASFF attributes, see [the section called “Other top-level attributes” \(p. 180\)](#).

```
"Findings": [  
  {  
    "Action (p. 188)": {  
      "ActionType": "string",  
      "AwsApiCallAction": {  
        "AffectedResources": {  
          "string": "string"  
        },  
        "Api": "string",  
        "CallerType": "string",  
        "DomainDetails": {  
          "Domain": "string"  
        },  
      },  
    },  
  ],  
]
```

```

"FirstSeen": "string",
"LastSeen": "string",
"RemoteIpDetails": {
  "City": {
    "CityName": "string"
  },
  "Country": {
    "CountryCode": "string",
    "CountryName": "string"
  },
  "IpAddressV4": "string",
  "Geolocation": {
    "Lat": "number",
    "Lon": "number"
  },
  "Organization": {
    "Asn": "number",
    "AsnOrg": "string",
    "Isp": "string",
    "Org": "string"
  }
},
"ServiceName": "string"
},
"DnsRequestAction": {
  "Blocked": "boolean",
  "Domain": "string",
  "Protocol": "string"
},
"NetworkConnectionAction": {
  "Blocked": "boolean",
  "ConnectionDirection": "string",
  "LocalPortDetails": {
    "Port": "number",
    "PortName": "string"
  },
  "Protocol": "string",
  "RemoteIpDetails": {
    "City": {
      "CityName": "string"
    },
    "Country": {
      "CountryCode": "string",
      "CountryName": "string"
    },
    "IpAddressV4": "string",
    "Geolocation": {
      "Lat": "number",
      "Lon": "number"
    },
    "Organization": {
      "Asn": "number",
      "AsnOrg": "string",
      "Isp": "string",
      "Org": "string"
    }
  },
  "RemotePortDetails": {
    "Port": "number",
    "PortName": "string"
  }
},
"PortProbeAction": {
  "Blocked": "boolean",
  "PortProbeDetails": [{
    "LocalIpDetails": {

```

```

        "IpAddressV4": "string"
    },
    "LocalPortDetails": {
        "Port": "number",
        "PortName": "string"
    },
    "RemoteIpDetails": {
        "City": {
            "CityName": "string"
        },
        "Country": {
            "CountryCode": "string",
            "CountryName": "string"
        },
        "GeoLocation": {
            "Lat": "number",
            "Lon": "number"
        },
        "IpAddressV4": "string",
        "Organization": {
            "Asn": "number",
            "AsnOrg": "string",
            "Isp": "string",
            "Org": "string"
        }
    }
}
}]
}
},
"AwsAccountId": "string",
"CompanyName": "string",
"Compliance (p. 189)": {
    "RelatedRequirements": ["string"],
    "Status": "string",
    "StatusReasons": [{
        "Description": "string",
        "ReasonCode": "string"
    }]
},
"Confidence": "number",
"CreatedAt": "string",
"Criticality": "number",
>Description": "string",
"FindingProviderFields (p. 189)": {
    "Confidence": "number",
    "Criticality": "number",
    "RelatedFindings": [{
        "ProductArn": "string",
        "Id": "string"
    }],
    "Severity": {
        "Label": "string",
        "Original": "string"
    },
    "Types": ["string"]
},
"FirstObservedAt": "string",
"GeneratorId": "string",
"Id": "string",
>LastObservedAt": "string",
"Malware (p. 189)": [{
    "Name": "string",
    "Path": "string",
    "State": "string",
    "Type": "string"
}],

```

```

"Network (p. 190)": {
  "DestinationDomain": "string",
  "DestinationIPv4": "string",
  "DestinationIPv6": "string",
  "DestinationPort": "number",
  "Direction": "string",
  "OpenPortRange": {
    "Begin": "integer",
    "End": "integer"
  },
  "Protocol": "string",
  "SourceDomain": "string",
  "SourceIPv4": "string",
  "SourceIPv6": "string",
  "SourceMac": "string",
  "SourcePort": "number"
},
"NetworkPath (p. 190)": [{
  "ComponentId": "string",
  "ComponentType": "string",
  "Egress": {
    "Destination": {
      "Address": ["string"],
      "PortRanges": [{
        "Begin": "integer",
        "End": "integer"
      }]
    }
  },
  "Protocol": "string",
  "Source": {
    "Address": ["string"],
    "PortRanges": [{
      "Begin": "integer",
      "End": "integer"
    }]
  }
}],
  "Ingress": {
    "Destination": {
      "Address": ["string"],
      "PortRanges": [{
        "Begin": "integer",
        "End": "integer"
      }]
    }
  },
  "Protocol": "string",
  "Source": {
    "Address": ["string"],
    "PortRanges": [{
      "Begin": "integer",
      "End": "integer"
    }]
  }
}],
"Note (p. 191)": {
  "Text": "string",
  "UpdatedAt": "string",
  "UpdatedBy": "string"
},
"PatchSummary (p. 191)": {
  "FailedCount": "number",
  "Id": "string",
  "InstalledCount": "number",
  "InstalledOtherCount": "number",
  "InstalledPendingReboot": "number",

```

```

    "InstalledRejectedCount": "number",
    "MissingCount": "number",
    "Operation": "string",
    "OperationEndTime": "string",
    "OperationStartTime": "string",
    "RebootOption": "string"
  },
  "Process (p. 192)": {
    "LaunchedAt": "string",
    "Name": "string",
    "ParentPid": "number",
    "Path": "string",
    "Pid": "number",
    "TerminatedAt": "string"
  },
  "ProductArn": "string",
  "ProductFields": {
    "string": "string"
  },
  "ProductName": "string",
  "RecordState": "string",
  "Region": "string",
  "RelatedFindings (p. 192)": [{
    "Id": "string",
    "ProductArn": "string"
  }],
  "Remediation (p. 192)": {
    "Recommendation": {
      "Text": "string",
      "Url": "string"
    }
  },
  "Resources (p. 193)": [{
    "DataClassification (p. 199)": {
      "DetailedResultsLocation": "string",
      "Result": {
        "AdditionalOccurrences": "boolean",
        "CustomDataIdentifiers": {
          "Detections": [{
            "Arn": "string",
            "Count": "integer",
            "Name": "string",
            "Occurrences": {
              "Cells": [{
                "CellReference": "string",
                "Column": "integer",
                "ColumnName": "string",
                "Row": "integer"
              }],
              "LineRanges": [{
                "End": "integer",
                "Start": "integer",
                "StartColumn": "integer"
              }],
              "OffsetRanges": [{
                "End": "integer",
                "Start": "integer",
                "StartColumn": "integer"
              }],
              "Pages": [{
                "LineRange": {
                  "End": "integer",
                  "Start": "integer",
                  "StartColumn": "integer"
                }
              },
              "OffsetRange": {

```



```

        "End": "integer",
        "Start": "integer",
        "StartColumn": "integer"
    },
    "PageNumber": "integer"
  }],
  "Records": [{
    "JsonPath": "string",
    "RecordIndex": "integer"
  }]
}
}],
"TotalCount": "integer"
},
"MimeType": "string",
"SensitiveData": [{
  "Category": "string",
  "Detections": [{
    "Count": "integer",
    "Occurrences": {
      "Cells": [{
        "CellReference": "string",
        "Column": "integer",
        "ColumnName": "string",
        "Row": "integer"
      }],
      "LineRanges": [{
        "End": "integer",
        "Start": "integer",
        "StartColumn": "integer"
      }],
      "OffsetRanges": [{
        "End": "integer",
        "Start": "integer",
        "StartColumn": "integer"
      }],
      "Pages": [{
        "LineRange": {
          "End": "integer",
          "Start": "integer",
          "StartColumn": "integer"
        },
        "OffsetRange": {
          "End": "integer",
          "Start": "integer",
          "StartColumn": "integer"
        },
        "PageNumber": "integer"
      }],
      "Records": [{
        "JsonPath": "string",
        "RecordIndex": "integer"
      }]
    }
  },
  "Type": "string"
}],
"TotalCount": "integer"
}],
"SizeClassified": "integer",
"Status": {
  "Code": "string",
  "Reason": "string"
}
}
},
"Details": {

```

```

"AwsApiGatewayRestApi (p. 208)": {
  "ApiKeySource": "string",
  "BinaryMediaTypes": ["string"],
  "CreateDate": "string",
  "Description": "string",
  "EndpointConfiguration": {
    "Types": ["string"]
  },
  "Id": "string",
  "MinimumCompressionSize": "number",
  "Name": "string",
  "Version": "string"
},
"AwsApiGatewayStage (p. 208)": {
  "AccessLogSettings": {
    "DestinationArn": "string",
    "Format": "string"
  },
  "CacheClusterEnabled": "boolean",
  "CacheClusterSize": "string",
  "CacheClusterStatus": "string",
  "CanarySettings": {
    "DeploymentId": "string",
    "PercentTraffic": "number",
    "StageVariableOverrides": [{
      "string": "string"
    }],
    "UseStageCache": "boolean"
  },
  "ClientCertificateId": "string",
  "CreateDate": "string",
  "DeploymentId": "string",
  "Description": "string",
  "DocumentationVersion": "string",
  "LastUpdatedDate": "string",
  "MethodSettings": [{
    "CacheDataEncrypted": "boolean",
    "CachingEnabled": "boolean",
    "CacheTtlInSeconds": "number",
    "DataTraceEnabled": "boolean",
    "HttpMethod": "string",
    "LoggingLevel": "string",
    "MetricsEnabled": "boolean",
    "RequireAuthorizationForCacheControl": "boolean",
    "ResourcePath": "string",
    "ThrottlingBurstLimit": "number",
    "ThrottlingRateLimit": "number",
    "UnauthorizedCacheControlHeaderStrategy": "string"
  }],
  "StageName": "string",
  "TracingEnabled": "boolean",
  "Variables": {
    "string": "string"
  },
  "WebAclArn": "string"
},
"AwsApiGatewayV2Api (p. 209)": {
  "ApiEndpoint": "string",
  "ApiId": "string",
  "ApiKeySelectionExpression": "string",
  "CorsConfiguration": {
    "AllowCredentials": "boolean",
    "AllowHeaders": ["string"],
    "AllowMethods": ["string"],
    "AllowOrigins": ["string"],
    "ExposeHeaders": ["string"],

```

```

    "MaxAge": "number"
  },
  "CreateDate": "string",
  "Description": "string",
  "Name": "string",
  "ProtocolType": "string",
  "RouteSelectionExpression": "string",
  "Version": "string"
},
"AwsApiGatewayV2Stage (p. 210)": {
  "AccessLogSettings": {
    "DestinationArn": "string",
    "Format": "string"
  },
  "ApiGatewayManaged": "boolean",
  "AutoDeploy": "boolean",
  "ClientCertificateId": "string",
  "CreateDate": "string",
  "DefaultRouteSettings": {
    "DataTraceEnabled": "boolean",
    "DetailedMetricsEnabled": "boolean",
    "LoggingLevel": "string",
    "ThrottlingBurstLimit": "number",
    "ThrottlingRateLimit": "number"
  },
  "DeploymentId": "string",
  "Description": "string",
  "LastDeploymentStatusMessage": "string",
  "LastUpdatedDate": "string",
  "RouteSettings": {
    "DetailedMetricsEnabled": "boolean",
    "LoggingLevel": "string",
    "DataTraceEnabled": "boolean",
    "ThrottlingBurstLimit": "number",
    "ThrottlingRateLimit": "number"
  },
  "StageName": "string",
  "StageVariables": [{
    "string": "string"
  }]
},
"AwsRdsDbSecurityGroup (p. 256)": {
  "DbSecurityGroupArn": "string",
  "DbSecurityGroupDescription": "string",
  "DbSecurityGroupName": "string",
  "Ec2SecurityGroups": [{
    "Ec2SecurityGroupOwnerId": "string",
    "Ec2SecurityGroupName": "string",
    "Ec2SecurityGroupOwnerId": "string",
    "Status": "string"
  }],
  "IpRanges": [{
    "CidrIp": "string",
    "Status": "string"
  }],
  "OwnerId": "string",
  "VpcId": "string"
},
"AwsAutoScalingAutoScalingGroup (p. 211)": {
  "AvailabilityZones": [{
    "Value": "string"
  }],
  "CreatedTime": "string",
  "HealthCheckGracePeriod": "integer",
  "HealthCheckType": "string",
  "LaunchConfigurationName": "string",

```

```
"LoadBalancerNames": ["string"],
"LaunchTemplate": {
    "LaunchTemplateId": "string",
    "LaunchTemplateName": "string",
    "Version": "string"
},
"MixedInstancesPolicy": {
    "InstancesDistribution": {
        "OnDemandAllocationStrategy": "string",
        "OnDemandBaseCapacity": "number",
        "OnDemandPercentageAboveBaseCapacity": "number",
        "SpotAllocationStrategy": "string",
        "SpotInstancePools": "number",
        "SpotMaxPrice": "string"
    },
    "LaunchTemplate": {
        "LaunchTemplateSpecification": {
            "LaunchTemplateId": "string",
            "LaunchTemplateName": "string",
            "Version": "string"
        },
        "CapacityRebalance": "boolean",
        "Overrides": [{
            "InstanceType": "string",
            "WeightedCapacity": "string"
        }]
    }
},
"AwsAutoScalingLaunchConfiguration (p. 212)": {
    "AssociatePublicIpAddress": "boolean",
    "BlockDeviceMappings": [{
        "DeviceName": "string",
        "Ebs": {
            "DeleteOnTermination": "boolean",
            "Encrypted": "boolean",
            "Iops": "number",
            "SnapshotId": "string",
            "VolumeSize": "number",
            "VolumeType": "string"
        },
        "NoDevice": "boolean",
        "VirtualName": "string"
    }],
    "ClassicLinkVpcId": "string",
    "ClassicLinkVpcSecurityGroups": ["string"],
    "CreatedTime": "string",
    "EbsOptimized": "boolean",
    "IamInstanceProfile": "string"
},
"ImageId": "string",
"InstanceMonitoring": {
    "Enabled": "boolean"
},
"InstanceType": "string",
"KernelId": "string",
"KeyName": "string",
"LaunchConfigurationName": "string",
"MetadataOptions": {
    "HttpEndPoint": "string",
    "HttpPutReponseHopLimit": "number",
    "HttpTokens": "string"
},
"PlacementTenancy": "string",
"RamdiskId": "string",
"SecurityGroups": ["string"],
```

```

    "SpotPrice": "string",
    "UserData": "string"
  },
  "AwsCertificateManagerCertificate (p. 213)": {
    "CertificateAuthorityArn": "string",
    "CreatedAt": "string",
    "DomainName": "string",
    "DomainValidationOptions": [{
      "DomainName": "string",
      "ResourceRecord": {
        "Name": "string",
        "Type": "string",
        "Value": "string"
      },
      "ValidationDomain": "string",
      "ValidationEmails": ["string"],
      "ValidationMethod": "string",
      "ValidationStatus": "string"
    }],
    "ExtendedKeyUsages": [{
      "Name": "string",
      "OID": "string"
    }],
    "FailureReason": "string",
    "ImportedAt": "string",
    "InUseBy": ["string"],
    "IssuedAt": "string",
    "Issuer": "string",
    "KeyAlgorithm": "string",
    "KeyUsages": [{
      "Name": "string"
    }],
    "NotAfter": "string",
    "NotBefore": "string",
    "Options": {
      "CertificateTransparencyLoggingPreference": "string"
    },
    "RenewalEligibility": "string",
    "RenewalSummary": {
      "DomainValidationOptions": [{
        "DomainName": "string",
        "ResourceRecord": {
          "Name": "string",
          "Type": "string",
          "Value": "string"
        },
        "ValidationDomain": "string",
        "ValidationEmails": ["string"],
        "ValidationMethod": "string",
        "ValidationStatus": "string"
      }],
      "RenewalStatus": "string",
      "RenewalStatusReason": "string",
      "UpdatedAt": "string"
    },
    "Serial": "string",
    "SignatureAlgorithm": "string",
    "Status": "string",
    "Subject": "string",
    "SubjectAlternativeNames": ["string"],
    "Type": "string"
  },
  "AwsCloudFormationStack (p. 215)": {
    "Capabilities": ["string"],
    "CreationTime": "string",
    "Description": "string",

```

```

"DisableRollback": "boolean",
"DriftInformation": {
  "StackDriftStatus": "string"
},
"EnableTerminationProtection": "boolean",
"LastUpdatedTime": "string",
"NotificationArns": ["string"],
"Outputs": [{
  "Description": "string",
  "OutputKey": "string",
  "OutputValue": "string"
}],
"RoleArn": "string",
"StackId": "string",
"StackName": "string",
"StackStatus": "string",
"StackStatusReason": "string",
"TimeoutInMinutes": "number"
},
"AwsCloudFrontDistribution (p. 215)": {
  "CacheBehaviors": {
    "Items": [{
      "ViewerProtocolPolicy": "string"
    }]
  },
  "DefaultCacheBehavior": {
    "ViewerProtocolPolicy": "string"
  },
  "DefaultRootObject": "string",
  "DomainName": "string",
  "Etag": "string",
  "LastModifiedTime": "string",
  "Logging": {
    "Bucket": "string",
    "Enabled": "boolean",
    "IncludeCookies": "boolean",
    "Prefix": "string"
  },
  "OriginGroups": {
    "Items": [{
      "FailoverCriteria": {
        "StatusCodes": {
          "Items": ["number"],
          "Quantity": "number"
        }
      }
    }]
  },
  "Origins": {
    "Items": [{
      "CustomOriginConfig": {
        "HttpPort": "number",
        "HttpsPort": "number",
        "OriginKeepaliveTimeout": "number",
        "OriginProtocolPolicy": "string",
        "OriginReadTimeout": "number",
        "OriginSslProtocols": {
          "Items": ["string"],
          "Quantity": "number"
        }
      }
    }],
    "DomainName": "string",
    "Id": "string",
    "OriginPath": "string",
    "S3OriginConfig": {
      "OriginAccessIdentity": "string"
    }
  }
}

```

```

    }
  }]
},
"Status": "string",
"ViewerCertificate": {
  "AcmCertificateArn": "string",
  "Certificate": "string",
  "CertificateSource": "string",
  "CloudFrontDefaultCertificate": "boolean",
  "IamCertificateId": "string",
  "MinimumProtocolVersion": "string",
  "SslSupportMethod": "string"
},
"WebAclId": "string"
},
"AwsCloudTrailTrail (p. 217)": {
  "CloudWatchLogsLogGroupArn": "string",
  "CloudWatchLogsRoleArn": "string",
  "HasCustomEventSelectors": "boolean",
  "HomeRegion": "string",
  "IncludeGlobalServiceEvents": "boolean",
  "IsMultiRegionTrail": "boolean",
  "IsOrganizationTrail": "boolean",
  "KmsKeyId": "string",
  "LogFileValidationEnabled": "boolean",
  "Name": "string",
  "S3BucketName": "string",
  "S3KeyPrefix": "string",
  "SnsTopicArn": "string",
  "SnsTopicName": "string",
  "TrailArn": "string"
},
"AwsCloudWatchAlarm (p. 217)": {
  "ActonsEnabled": "boolean",
  "AlarmActions": ["string"],
  "AlarmArn": "string",
  "AlarmConfigurationUpdatedTimestamp": "string",
  "AlarmDescription": "string",
  "AlarmName": "string",
  "ComparisonOperator": "string",
  "DatapointsToAlarm": "number",
  "Dimensions": [{
    "Name": "string",
    "Value": "string"
  }],
  "EvaluateLowSampleCountPercentile": "string",
  "EvaluationPeriods": "number",
  "ExtendedStatistic": "string",
  "InsufficientDataActions": ["string"],
  "MetricName": "string",
  "Namespace": "string",
  "OkActions": ["string"],
  "Period": "number",
  "Statistic": "string",
  "Threshold": "number",
  "ThresholdMetricId": "string",
  "TreatMissingData": "string",
  "Unit": "string"
},
"AwsCodeBuildProject (p. 218)": {
  "Artifacts": [{
    "ArtifactIdentifier": "string",
    "EncryptionDisabled": "boolean",
    "Location": "string",
    "Name": "string",
    "NamespaceType": "string",

```

```

    "OverrideArtifactName": "boolean",
    "Packaging": "string",
    "Path": "string",
    "Type": "string"
  }],
  "SecondaryArtifacts": [{
    "ArtifactIdentifier": "string",
    "Type": "string",
    "Location": "string",
    "Name": "string",
    "NamespaceType": "string",
    "Packaging": "string",
    "Path": "string",
    "EncryptionDisabled": "boolean",
    "OverrideArtifactName": "boolean"
  }],
  "EncryptionKey": "string",
  "Certificate": "string",
  "Environment": {
    "Certificate": "string",
    "EnvironmentVariables": [{
      "Name": "string",
      "Type": "string",
      "Value": "string"
    }],
    "ImagePullCredentialsType": "string",
    "PrivilegedMode": "boolean",
    "RegistryCredential": {
      "Credential": "string",
      "CredentialProvider": "string"
    },
    "Type": "string"
  },
  "LogsConfig": {
    "CloudWatchLogs": {
      "GroupName": "string",
      "Status": "string",
      "StreamName": "string"
    },
    "S3Logs": {
      "EncryptionDisabled": "boolean",
      "Location": "string",
      "Status": "string"
    }
  },
  "Name": "string",
  "ServiceRole": "string",
  "Source": {
    "Type": "string",
    "Location": "string",
    "GitCloneDepth": "integer"
  },
  "VpcConfig": {
    "VpcId": "string",
    "Subnets": ["string"],
    "SecurityGroupIds": ["string"]
  }
},
"AwsDynamoDbTable (p. 219)": {
  "AttributeDefinitions": [{
    "AttributeName": "string",
    "AttributeType": "string"
  }],
  "BillingModeSummary": {
    "BillingMode": "string",
    "LastUpdateToPayPerRequestDateTime": "string"
  }
}

```



```

},
"CreationDateTime": "string",
"GlobalSecondaryIndexes": [{
  "Backfilling": "boolean",
  "IndexArn": "string",
  "IndexName": "string",
  "IndexSizeBytes": "number",
  "IndexStatus": "string",
  "ItemCount": "number",
  "KeySchema": [{
    "AttributeName": "string",
    "KeyType": "string"
  }],
}],
"Projection": {
  "NonKeyAttributes": ["string"],
  "ProjectionType": "string"
},
"ProvisionedThroughput": {
  "LastDecreaseDateTime": "string",
  "LastIncreaseDateTime": "string",
  "NumberOfDecreasesToday": "number",
  "ReadCapacityUnits": "number",
  "WriteCapacityUnits": "number"
}
}],
"GlobalTableVersion": "string",
"ItemCount": "number",
"KeySchema": [{
  "AttributeName": "string",
  "KeyType": "string"
}],
"LatestStreamArn": "string",
"LatestStreamLabel": "string",
"LocalSecondaryIndexes": [{
  "IndexArn": "string",
  "IndexName": "string",
  "KeySchema": [{
    "AttributeName": "string",
    "KeyType": "string"
  }],
  "Projection": {
    "NonKeyAttributes": ["string"],
    "ProjectionType": "string"
  }
}],
"ProvisionedThroughput": {
  "LastDecreaseDateTime": "string",
  "LastIncreaseDateTime": "string",
  "NumberOfDecreasesToday": "number",
  "ReadCapacityUnits": "number",
  "WriteCapacityUnits": "number"
},
"Replicas": [{
  "GlobalSecondaryIndexes": [{
    "IndexName": "string",
    "ProvisionedThroughputOverride": {
      "ReadCapacityUnits": "number"
    }
  }],
  "KmsMasterKeyId": "string",
  "ProvisionedThroughputOverride": {
    "ReadCapacityUnits": "number"
  }
},
"RegionName": "string",
"ReplicaStatus": "string",
"ReplicaStatusDescription": "string"

```

```

    }],
    "RestoreSummary": {
      "RestoreDateTime": "string",
      "RestoreInProgress": "boolean",
      "SourceBackupArn": "string",
      "SourceTableArn": "string"
    },
    "SseDescription": {
      "InaccessibleEncryptionDateTime": "string",
      "KmsMasterKeyArn": "string",
      "SseType": "string",
      "Status": "string"
    },
    "StreamSpecification": {
      "StreamEnabled": "boolean",
      "StreamViewType": "string"
    },
    "TableId": "string",
    "TableName": "string",
    "TableSizeBytes": "number",
    "TableStatus": "string"
  },
  "AwsEc2Eip (p. 222)": {
    "AllocationId": "string",
    "AssociationId": "string",
    "Domain": "string",
    "InstanceId": "string",
    "NetworkBorderGroup": "string",
    "NetworkInterfaceId": "string",
    "NetworkInterfaceOwnerId": "string",
    "PrivateIpAddress": "string",
    "PublicIp": "string",
    "PublicIpv4Pool": "string"
  },
  "AwsEc2Instance (p. 222)": {
    "IamInstanceProfileArn": "string",
    "ImageId": "string",
    "Ipv4Addresses": ["string"],
    "Ipv6Addresses": ["string"],
    "KeyName": "string",
    "LaunchedAt": "string",
    "MetadataOptions": {
      "HttpEndpoint": "string",
      "HttpProtocolIpv6": "string",
      "HttpPutResponseHopLimit": "number",
      "HttpTokens": "string",
      "InstanceMetadataTags": "string",
    },
    "NetworkInterfaces": [{
      "NetworkInterfaceId": "string"
    }],
    "SubnetId": "string",
    "Type": "string",
    "VirtualizationType": "string",
    "VpcId": "string"
  },
  "AwsEc2NetworkAcl (p. 223)": {
    "Associations": [{
      "NetworkAclAssociationId": "string",
      "NetworkAclId": "string",
      "SubnetId": "string"
    }],
    "Entries": [{
      "CidrBlock": "string",
      "Egress": "boolean",
      "IcmpTypeCode": {

```

```

        "Code": "number",
        "Type": "number"
    },
    "Ipv6CidrBlock": "string",
    "PortRange": {
        "From": "number",
        "To": "number"
    },
    },
    "Protocol": "string",
    "RuleAction": "string",
    "RuleNumber": "number"
}],
    "IsDefault": "boolean",
    "NetworkAclId": "string",
    "OwnerId": "string",
    "VpcId": "string"
},
"AwsEc2NetworkInterface (p. 223)": {
    "Attachment": {
        "AttachmentId": "string",
        "AttachTime": "string",
        "DeleteOnTermination": "boolean",
        "DeviceIndex": "number",
        "InstanceId": "string",
        "InstanceOwnerId": "string",
        "Status": "string"
    },
    "Ipv6Addresses": [{
        "Ipv6Address": "string"
    }],
    "NetworkInterfaceId": "string",
    "PrivateIpAddresses": [{
        "PrivateDnsName": "string",
        "PrivateIpAddress": "string"
    }],
    "PublicDnsName": "string",
    "PublicIp": "string",
    "SecurityGroups": [{
        "GroupId": "string",
        "GroupName": "string"
    }],
    "SourceDestCheck": "boolean"
},
"AwsEc2SecurityGroup (p. 224)": {
    "GroupId": "string",
    "GroupName": "string",
    "IpPermissions": [{
        "FromPort": "number",
        "IpProtocol": "string",
        "IpRanges": [{
            "CidrIp": "string"
        }],
        "Ipv6Ranges": [{
            "CidrIpv6": "string"
        }],
        "PrefixListIds": [{
            "PrefixListId": "string"
        }],
        "ToPort": "number",
        "UserIdGroupPairs": [{
            "GroupId": "string",
            "GroupName": "string",
            "PeeringStatus": "string",
            "UserId": "string",
            "VpcId": "string",
            "VpcPeeringConnectionId": "string"
        }],
    }],
    "PeeringStatus": "string",
    "UserId": "string",
    "VpcId": "string",
    "VpcPeeringConnectionId": "string"
}

```

```

    ]]
  ]],
  "IpPermissionsEgress": [{
    "FromPort": "number",
    "IpProtocol": "string",
    "IpRanges": [{
      "CidrIp": "string"
    }],
    "Ipv6Ranges": [{
      "CidrIpv6": "string"
    }],
    "PrefixListIds": [{
      "PrefixListId": "string"
    }],
    "ToPort": "number",
    "UserIdGroupPairs": [{
      "GroupId": "string",
      "GroupName": "string",
      "PeeringStatus": "string",
      "UserId": "string",
      "VpcId": "string",
      "VpcPeeringConnectionId": "string"
    }
  ]],
  "OwnerId": "string",
  "VpcId": "string"
},
"AwsEc2Subnet (p. 224)": {
  "AssignIpv6AddressOnCreation": "boolean",
  "AvailabilityZone": "string",
  "AvailabilityZoneId": "string",
  "AvailableIpAddressCount": "number",
  "CidrBlock": "string",
  "DefaultForAz": "boolean",
  "Ipv6CidrBlockAssociationSet": [{
    "AssociationId": "string",
    "Ipv6CidrBlock": "string",
    "CidrBlockState": "string"
  }],
  "MapPublicIpOnLaunch": "boolean",
  "OwnerId": "string",
  "State": "string",
  "SubnetArn": "string",
  "SubnetId": "string",
  "VpcId": "string"
},
"AwsEc2TransitGateway (p. 225)": {
  "AmazonSideAsn": "number",
  "AssociationDefaultRouteTableId": "string",
  "AutoAcceptSharedAttachments": "string",
  "DefaultRouteTableAssociation": "string",
  "DefaultRouteTablePropagation": "string",
  "Description": "string",
  "DnsSupport": "string",
  "Id": "string",
  "MulticastSupport": "string",
  "PropagationDefaultRouteTableId": "string",
  "TransitGatewayCidrBlocks": ["string"],
  "VpnEcmpSupport": "string"
},
"AwsEc2Volume (p. 225)": {
  "Attachments": [{
    "AttachTime": "string",
    "DeleteOnTermination": "boolean",
    "InstanceId": "string",
    "Status": "string"
  }],

```

```

    }],
    "CreateTime": "string",
    "DeviceName": "string",
    "Encrypted": "boolean",
    "KmsKeyId": "string",
    "Size": "number",
    "SnapshotId": "string",
    "Status": "string",
    "VolumeId": "string",
    "VolumeScanStatus": "string",
    "VolumeType": "string"
  },
  "AwsEc2Vpc (p. 226)": {
    "CidrBlockAssociationSet": [{
      "AssociationId": "string",
      "CidrBlock": "string",
      "CidrBlockState": "string"
    }],
    "DhcpOptionsId": "string",
    "Ipv6CidrBlockAssociationSet": [{
      "AssociationId": "string",
      "CidrBlockState": "string",
      "Ipv6CidrBlock": "string"
    }],
    "State": "string"
  },
  "AwsEc2VpcEndpointService (p. 227)": {
    "AcceptanceRequired": "boolean",
    "AvailabilityZones": ["string"],
    "BaseEndpointDnsNames": ["string"],
    "ManagesVpcEndpoints": "boolean",
    "GatewayLoadBalancerArns": ["string"],
    "NetworkLoadBalancerArns": ["string"],
    "PrivateDnsName": "string",
    "ServiceId": "string",
    "ServiceName": "string",
    "ServiceState": "string",
    "ServiceType": [{
      "ServiceType": "string"
    }]
  },
  "AwsEc2VpcPeeringConnection (p. 227)": {
    "AccepterVpcInfo": {
      "CidrBlock": "string",
      "CidrBlockSet": [{
        "CidrBlock": "string"
      }],
      "Ipv6CidrBlockSet": [{
        "Ipv6CidrBlock": "string"
      }],
      "OwnerId": "string",
      "PeeringOptions": {
        "AllowDnsResolutionFromRemoteVpc": "boolean",
        "AllowEgressFromLocalClassicLinkToRemoteVpc": "boolean",
        "AllowEgressFromLocalVpcToRemoteClassicLink": "boolean"
      },
      "Region": "string",
      "VpcId": "string"
    },
    "ExpirationTime": "string",
    "RequesterVpcInfo": {
      "CidrBlock": "string",
      "CidrBlockSet": [{
        "CidrBlock": "string"
      }],
      "Ipv6CidrBlockSet": [{

```

```

    "Ipv6CidrBlock": "string"
  }],
  "OwnerId": "string",
  "PeeringOptions": {
    "AllowDnsResolutionFromRemoteVpc": "boolean",
    "AllowEgressFromLocalClassicLinkToRemoteVpc": "boolean",
    "AllowEgressFromLocalVpcToRemoteClassicLink": "boolean"
  },
  "Region": "string",
  "VpcId": "string"
},
"Status": {
  "Code": "string",
  "Message": "string"
},
"VpcPeeringConnectionId": "string"
},
"AwsEc2VpnConnection (p. 228)": {
  "Category": "string",
  "CustomerGatewayConfiguration": "string",
  "CustomerGatewayId": "string",
  "Options": {
    "StaticRoutesOnly": "boolean",
    "TunnelOptions": [{
      "DpdTimeoutSeconds": "number",
      "IkeVersions": ["string"],
      "OutsideIpAddress": "string",
      "Phase1DhGroupNumbers": ["number"],
      "Phase1EncryptionAlgorithms": ["string"],
      "Phase1IntegrityAlgorithms": ["string"],
      "Phase1LifetimeSeconds": "number",
      "Phase2DhGroupNumbers": ["number"],
      "Phase2EncryptionAlgorithms": ["string"],
      "Phase2IntegrityAlgorithms": ["string"],
      "Phase2LifetimeSeconds": "number",
      "PreSharedKey": "string",
      "RekeyFuzzPercentage": "number",
      "RekeyMarginTimeSeconds": "number",
      "ReplayWindowSize": "number",
      "TunnelInsideCidr": "string"
    }]
  },
  "Routes": [{
    "DestinationCidrBlock": "string",
    "State": "string"
  }],
  "State": "string",
  "TransitGatewayId": "string",
  "Type": "string",
  "VgwTelemetry": [{
    "AcceptedRouteCount": "number",
    "CertificateArn": "string",
    "LastStatusChange": "string",
    "OutsideIpAddress": "string",
    "Status": "string",
    "StatusMessage": "string"
  }],
  "VpnConnectionId": "string",
  "VpnGatewayId": "string"
},
"AwsEcrContainerImage (p. 229)": {
  "Architecture": "string",
  "ImageDigest": "string",
  "ImagePublishedAt": "string",
  "ImageTags": ["string"],
  "RegistryId": "string",

```

```

    "RepositoryName": "string"
  },
  "AwsEcrRepository (p. 230)": {
    "Arn": "string",
    "ImageScanningConfiguration": {
      "ScanOnPush": "boolean"
    },
    "ImageTagMutability": "string",
    "LifecyclePolicy": {
      "LifecyclePolicyText": "string",
      "RegistryId": "string"
    },
    "RepositoryName": "string",
    "RepositoryPolicyText": "string"
  },
  "AwsEcsCluster (p. 230)": {
    "ActiveServicesCount": "number",
    "CapacityProviders": ["string"],
    "ClusterArn": "string",
    "ClusterName": "string",
    "ClusterSettings": [{
      "Name": "string",
      "Value": "string"
    }],
    "Configuration": {
      "ExecuteCommandConfiguration": {
        "KmsKeyId": "string",
        "LogConfiguration": {
          "CloudWatchEncryptionEnabled": "boolean",
          "CloudWatchLogGroupName": "string",
          "S3BucketName": "string",
          "S3EncryptionEnabled": "boolean",
          "S3KeyPrefix": "string"
        },
        "Logging": "string"
      }
    },
    "DefaultCapacityProviderStrategy": [{
      "Base": "number",
      "CapacityProvider": "string",
      "Weight": "number"
    }],
    "RegisteredContainerInstancesCount": "number",
    "RunningTasksCount": "number",
    "Status": "string"
  },
  "AwsEcsContainer (p. 231)": {
    "Image": "string",
    "MountPoints": [{
      "ContainerPath": "string",
      "SourceVolume": "string"
    }],
    "Name": "string",
    "Privileged": "boolean"
  },
  "AwsEcsService (p. 232)": {
    "CapacityProviderStrategy": [{
      "Base": "number",
      "CapacityProvider": "string",
      "Weight": "number"
    }],
    "Cluster": "string",
    "DeploymentConfiguration": {
      "DeploymentCircuitBreaker": {
        "Enable": "boolean",
        "Rollback": "boolean"
      }
    }
  }
}

```

```

    },
    "MaximumPercent": "number",
    "MinimumHealthyPercent": "number"
  },
  "DeploymentController": {
    "Type": "string"
  },
},
"DesiredCount": "number",
"EnableEcsManagedTags": "boolean",
"EnableExecuteCommand": "boolean",
"HealthCheckGracePeriodSeconds": "number",
"LaunchType": "string",
"LoadBalancers": [{
  "ContainerName": "string",
  "ContainerPort": "number",
  "LoadBalancerName": "string",
  "TargetGroupArn": "string"
}],
"Name": "string",
"NetworkConfiguration": {
  "AwsVpcConfiguration": {
    "AssignPublicIp": "string",
    "SecurityGroups": ["string"],
    "Subnets": ["string"]
  }
},
"PlacementConstraints": [{
  "Expression": "string",
  "Type": "string"
}],
"PlacementStrategies": [{
  "Field": "string",
  "Type": "string"
}],
"PlatformVersion": "string",
"PropagateTags": "string",
"Role": "string",
"SchedulingStrategy": "string",
"ServiceArn": "string",
"ServiceName": "string",
"ServiceRegistries": [{
  "ContainerName": "string",
  "ContainerPort": "number",
  "Port": "number",
  "RegistryArn": "string"
}],
"TaskDefinition": "string"
},
"AwsEcsTask (p. 233)": {
  "CreatedAt": "string",
  "ClusterArn": "string",
  "Group": "string",
  "StartedAt": "string",
  "StartedBy": "string",
  "TaskDefinitionArn": "string",
  "Version": "number",
  "Volumes": [{
    "Name": "string",
    "Host": {
      "SourcePath": "string"
    }
  ]
},
"Containers": [{
  "Image": "string",
  "MountPoints": [{
    "ContainerPath": "string",

```



```

    "SourceVolume": "string"
  }],
  "Name": "string",
  "Privileged": "boolean"
}]
},
"AwsEcsTaskDefinition (p. 234)": {
  "ContainerDefinitions": [{
    "Command": ["string"],
    "Cpu": "number",
    "DependsOn": [{
      "Condition": "string",
      "ContainerName": "string"
    }],
    "DisableNetworking": "boolean",
    "DnsSearchDomains": ["string"],
    "DnsServers": ["string"],
    "DockerLabels": {
      "string": "string"
    },
    "DockerSecurityOptions": ["string"],
    "EntryPoint": ["string"],
    "Environment": [{
      "Name": "string",
      "Value": "string"
    }],
    "EnvironmentFiles": [{
      "Type": "string",
      "Value": "string"
    }],
    "Essential": "boolean",
    "ExtraHosts": [{
      "Hostname": "string",
      "IpAddress": "string"
    }],
    "FirelensConfiguration": {
      "Options": {
        "string": "string"
      },
      "Type": "string"
    },
    "HealthCheck": {
      "Command": ["string"],
      "Interval": "number",
      "Retries": "number",
      "StartPeriod": "number",
      "Timeout": "number"
    },
    "Hostname": "string",
    "Image": "string",
    "Interactive": "boolean",
    "Links": ["string"],
    "LinuxParameters": {
      "Capabilities": {
        "Add": ["string"],
        "Drop": ["string"]
      },
      "Devices": [{
        "ContainerPath": "string",
        "HostPath": "string",
        "Permissions": ["string"]
      }],
      "InitProcessEnabled": "boolean",
      "MaxSwap": "number",
      "SharedMemorySize": "number",
      "Swappiness": "number",

```

```

    "Tmpfs": [{
      "ContainerPath": "string",
      "MountOptions": ["string"],
      "Size": "number"
    }]
  },
  "LogConfiguration": {
    "LogDriver": "string",
    "Options": {
      "string": "string"
    },
    "SecretOptions": [{
      "Name": "string",
      "ValueFrom": "string"
    }]
  },
  "Memory": "number",
  "MemoryReservation": "number",
  "MountPoints": [{
    "ContainerPath": "string",
    "ReadOnly": "boolean",
    "SourceVolume": "string"
  }],
  "Name": "string",
  "PortMappings": [{
    "ContainerPort": "number",
    "HostPort": "number",
    "Protocol": "string"
  }],
  "Privileged": "boolean",
  "PseudoTerminal": "boolean",
  "ReadOnlyRootFilesystem": "boolean",
  "RepositoryCredentials": {
    "CredentialsParameter": "string"
  },
  "ResourceRequirements": [{
    "Type": "string",
    "Value": "string"
  }],
  "Secrets": [{
    "Name": "string",
    "ValueFrom": "string"
  }],
  "StartTimeout": "number",
  "StopTimeout": "number",
  "SystemControls": [{
    "Namespace": "string",
    "Value": "string"
  }],
  "Ulimits": [{
    "HardLimit": "number",
    "Name": "string",
    "SoftLimit": "number"
  }],
  "User": "string",
  "VolumesFrom": [{
    "ReadOnly": "boolean",
    "SourceContainer": "string"
  }],
  "WorkingDirectory": "string"
}],
"Cpu": "string",
"ExecutionRoleArn": "string",
"Family": "string",
"InferenceAccelerators": [{
  "DeviceName": "string",

```

```

    "DeviceType": "string"
  }],
  "IpcMode": "string",
  "Memory": "string",
  "NetworkMode": "string",
  "PidMode": "string",
  "PlacementConstraints": [{
    "Expression": "string",
    "Type": "string"
  }],
  "ProxyConfiguration": {
    "ContainerName": "string",
    "ProxyConfigurationProperties": [{
      "Name": "string",
      "Value": "string"
    }],
    "Type": "string"
  },
  "RequiresCompatibilities": ["string"],
  "TaskRoleArn": "string",
  "Status": "string",
  "Volumes": [{
    "DockerVolumeConfiguration": {
      "Autoprovision": "boolean",
      "Driver": "string",
      "DriverOpts": {
        "string": "string"
      },
      "Labels": {
        "string": "string"
      },
      "Scope": "string"
    },
    "EfsVolumeConfiguration": {
      "AuthorizationConfig": {
        "AccessPointId": "string",
        "Iam": "string"
      },
      "FilesystemId": "string",
      "RootDirectory": "string",
      "TransitEncryption": "string",
      "TransitEncryptionPort": "number"
    },
    "Host": {
      "SourcePath": "string"
    },
    "Name": "string"
  }],
  "AwsEfsAccessPoint (p. 236)": {
    "AccessPointId": "string",
    "Arn": "string",
    "ClientToken": "string",
    "FilesystemId": "string",
    "PosixUser": {
      "Gid": "string",
      "SecondaryGids": ["string"],
      "Uid": "string"
    },
    "RootDirectory": {
      "CreationInfo": {
        "OwnerGid": "string",
        "OwnerUid": "string",
        "Permissions": "string"
      },
      "Path": "string"
    }
  }
}

```

```

    }
  },
  "AwsEksCluster (p. 237)": {
    "Arn": "string",
    "CertificateAuthorityData": "string",
    "ClusterStatus": "string",
    "Endpoint": "string",
    "Logging": {
      "ClusterLogging": [{
        "Enabled": "boolean",
        "Types": ["string"]
      }]
    },
    "Name": "string",
    "ResourcesVpcConfig": {
      "SecurityGroupIds": ["string"],
      "SubnetIds": ["string"]
    },
    "RoleArn": "string",
    "Version": "string"
  },
  "AwsElasticBeanstalkEnvironment (p. 238)": {
    "ApplicationName": "string",
    "Cname": "string",
    "DateCreated": "string",
    "DateUpdated": "string",
    "Description": "string",
    "EndpointUrl": "string",
    "EnvironmentArn": "string",
    "EnvironmentId": "string",
    "EnvironmentLinks": [{
      "EnvironmentName": "string",
      "LinkName": "string"
    }],
    "EnvironmentName": "string",
    "OptionSettings": [{
      "Namespace": "string",
      "OptionName": "string",
      "ResourceName": "string",
      "Value": "string"
    }],
    "PlatformArn": "string",
    "SolutionStackName": "string",
    "Status": "string",
    "Tier": {
      "Name": "string",
      "Type": "string",
      "Version": "string"
    },
    "VersionLabel": "string"
  },
  "AwsElasticSearchDomain (p. 239)": {
    "AccessPolicies": "string",
    "DomainStatus": {
      "DomainId": "string",
      "DomainName": "string",
      "Endpoint": "string",
      "Endpoints": {
        "string": "string"
      }
    },
    "DomainEndpointOptions": {
      "EnforceHTTPS": "boolean",
      "TLSSecurityPolicy": "string"
    },
    "ElasticsearchClusterConfig": {

```

```

    "DedicatedMasterCount": "number",
    "DedicatedMasterEnabled": "boolean",
    "DedicatedMasterType": "string",
    "InstanceCount": "number",
    "InstanceType": "string",
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": "number"
    },
    "ZoneAwarenessEnabled": "boolean"
  },
  "ElasticsearchVersion": "string",
  "EncryptionAtRestOptions": {
    "Enabled": "boolean",
    "KmsKeyId": "string"
  },
  "LogPublishingOptions": {
    "AuditLogs": {
      "CloudWatchLogsLogGroupArn": "string",
      "Enabled": "boolean"
    },
    "IndexSlowLogs": {
      "CloudWatchLogsLogGroupArn": "string",
      "Enabled": "boolean"
    },
    "SearchSlowLogs": {
      "CloudWatchLogsLogGroupArn": "string",
      "Enabled": "boolean"
    }
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": "boolean"
  },
  "ServiceSoftwareOptions": {
    "AutomatedUpdateDate": "string",
    "Cancellable": "boolean",
    "CurrentVersion": "string",
    "Description": "string",
    "NewVersion": "string",
    "UpdateAvailable": "boolean",
    "UpdateStatus": "string"
  },
  "VPCOptions": {
    "AvailabilityZones": [
      "string"
    ],
    "SecurityGroupIds": [
      "string"
    ],
    "SubnetIds": [
      "string"
    ],
    "VPCId": "string"
  }
},
"AwsElbLoadBalancer (p. 240)": {
  "AvailabilityZones": ["string"],
  "BackendServerDescriptions": [{
    "InstancePort": "number",
    "PolicyNames": ["string"]
  }],
  "CanonicalHostedZoneName": "string",
  "CanonicalHostedZoneNameID": "string",
  "CreatedTime": "string",
  "DnsName": "string",
  "HealthCheck": {
    "HealthyThreshold": "number",

```

```

    "Interval": "number",
    "Target": "string",
    "Timeout": "number",
    "UnhealthyThreshold": "number"
  },
  "Instances": [{
    "InstanceId": "string"
  }],
  "ListenerDescriptions": [{
    "Listener": {
      "InstancePort": "number",
      "InstanceProtocol": "string",
      "LoadBalancerPort": "number",
      "Protocol": "string",
      "SslCertificateId": "string"
    },
    "PolicyNames": ["string"]
  }],
  "LoadBalancerAttributes": {
    "AccessLog": {
      "EmitInterval": "number",
      "Enabled": "boolean",
      "S3BucketName": "string",
      "S3BucketPrefix": "string"
    },
    "ConnectionDraining": {
      "Enabled": "boolean",
      "Timeout": "number"
    },
    "ConnectionSettings": {
      "IdleTimeout": "number"
    },
    "CrossZoneLoadBalancing": {
      "Enabled": "boolean"
    },
    "AdditionalAttributes": [{
      "Key": "string",
      "Value": "string"
    }]
  },
  "LoadBalancerName": "string",
  "Policies": {
    "AppCookieStickinessPolicies": [{
      "CookieName": "string",
      "PolicyName": "string"
    }],
    "LbCookieStickinessPolicies": [{
      "CookieExpirationPeriod": "number",
      "PolicyName": "string"
    }],
    "OtherPolicies": ["string"]
  },
  "Scheme": "string",
  "SecurityGroups": ["string"],
  "SourceSecurityGroup": {
    "GroupName": "string",
    "OwnerAlias": "string"
  },
  "Subnets": ["string"],
  "VpcId": "string"
},
"AwsElbv2LoadBalancer (p. 242)": {
  "AvailabilityZones": {
    "SubnetId": "string",
    "ZoneName": "string"
  }
},

```

```

"CanonicalHostedZoneId": "string",
"CreatedTime": "string",
"DNSName": "string",
"IpAddressType": "string",
"LoadBalancerAttributes": [{
  "Key": "string",
  "Value": "string"
}],
"Scheme": "string",
"SecurityGroups": ["string"],
"State": {
  "Code": "string",
  "Reason": "string"
},
"Type": "string",
"VpcId": "string"
},
"AwsIamAccessKey (p. 243)": {
  "AccessKeyId": "string",
  "AccountId": "string",
  "CreatedAt": "string",
  "PrincipalId": "string",
  "PrincipalName": "string",
  "PrincipalType": "string",
  "SessionContext": {
    "Attributes": {
      "CreationDate": "string",
      "MfaAuthenticated": "boolean"
    },
    "SessionIssuer": {
      "AccountId": "string",
      "Arn": "string",
      "PrincipalId": "string",
      "Type": "string",
      "UserName": "string"
    }
  },
  "Status": "string"
},
"AwsIamGroup (p. 243)": {
  "AttachedManagedPolicies": [{
    "PolicyArn": "string",
    "PolicyName": "string"
  }],
  "CreateDate": "string",
  "GroupId": "string",
  "GroupName": "string",
  "GroupPolicyList": [{
    "PolicyName": "string"
  }],
  "Path": "string"
},
"AwsIamPolicy (p. 244)": {
  "AttachmentCount": "number",
  "CreateDate": "string",
  "DefaultVersionId": "string",
  "Description": "string",
  "IsAttachable": "boolean",
  "Path": "string",
  "PermissionsBoundaryUsageCount": "number",
  "PolicyId": "string",
  "PolicyName": "string",
  "PolicyVersionList": [{
    "CreateDate": "string",
    "IsDefaultVersion": "boolean",
    "VersionId": "string"
  }],

```

```

    }],
    "UpdateDate": "string"
  },
  "AwsIamRole (p. 244)": {
    "AssumeRolePolicyDocument": "string",
    "AttachedManagedPolicies": [{
      "PolicyArn": "string",
      "PolicyName": "string"
    }],
    "CreateDate": "string",
    "InstanceProfileList": [{
      "Arn": "string",
      "CreateDate": "string",
      "InstanceProfileId": "string",
      "InstanceProfileName": "string",
      "Path": "string",
      "Roles": [{
        "Arn": "string",
        "AssumeRolePolicyDocument": "string",
        "CreateDate": "string",
        "Path": "string",
        "RoleId": "string",
        "RoleName": "string"
      }]
    }],
    "MaxSessionDuration": "number",
    "Path": "string",
    "PermissionsBoundary": {
      "PermissionsBoundaryArn": "string",
      "PermissionsBoundaryType": "string"
    },
    "RoleId": "string",
    "RoleName": "string",
    "RolePolicyList": [{
      "PolicyName": "string"
    }]
  },
  "AwsIamUser (p. 245)": {
    "AttachedManagedPolicies": [{
      "PolicyArn": "string",
      "PolicyName": "string"
    }],
    "CreateDate": "string",
    "GroupList": ["string"],
    "Path": "string",
    "PermissionsBoundary": {
      "PermissionsBoundaryArn": "string",
      "PermissionsBoundaryType": "string"
    },
    "UserId": "string",
    "UserName": "string",
    "UserPolicyList": [{
      "PolicyName": "string"
    }]
  },
  "AwsKinesisStream (p. 246)": {
    "Arn": "string",
    "Name": "string",
    "RetentionPeriodHours": "number",
    "ShardCount": "number",
    "StreamEncryption": {
      "EncryptionType": "string",
      "KeyId": "string"
    }
  },
  "AwsKmsKey (p. 246)": {

```



```

    "AWSAccountId": "string",
    "CreationDate": "string",
    "Description": "string",
    "KeyId": "string",
    "KeyManager": "string",
    "KeyRotationStatus": "boolean",
    "KeyState": "string",
    "Origin": "string"
  },
  "AwsLambdaFunction (p. 246)": {
    "Code": {
      "S3Bucket": "string",
      "S3Key": "string",
      "S3ObjectVersion": "string",
      "ZipFile": "string"
    },
    "CodeSha256": "string",
    "DeadLetterConfig": {
      "TargetArn": "string"
    },
    "Environment": {
      "Variables": {
        "string": "string"
      },
      "Error": {
        "ErrorCode": "string",
        "Message": "string"
      }
    },
    "FunctionName": "string",
    "Handler": "string",
    "KmsKeyArn": "string",
    "LastModified": "string",
    "Layers": {
      "Arn": "string",
      "CodeSize": "number"
    },
    "RevisionId": "string",
    "Role": "string",
    "Runtime": "string",
    "Timeout": "integer",
    "TracingConfig": {
      "Mode": "string"
    },
    "Version": "string",
    "VpcConfig": {
      "SecurityGroupIds": ["string"],
      "SubnetIds": ["string"]
    },
    "MasterArn": "string",
    "MemorySize": "number"
  },
  "AwsLambdaLayerVersion (p. 247)": {
    "CompatibleRuntimes": [
      "string"
    ],
    "CreateDate": "string",
    "Version": "number"
  },
  "AwsNetworkFirewallFirewall (p. 248)": {
    "DeleteProtection": "boolean",
    "Description": "string",
    "FirewallArn": "string",
    "FirewallId": "string",
    "FirewallName": "string",
    "FirewallPolicyArn": "string",

```

```

"FirewallPolicyChangeProtection": "boolean",
"SubnetChangeProtection": "boolean",
"SubnetMappings": [{
  "SubnetId": "string"
}],
"VpcId": "string"
},
"AwsNetworkFirewallFirewallPolicy (p. 248)": {
  "Description": "string",
  "FirewallPolicy": {
    "StatefulRuleGroupReferences": [{
      "ResourceArn": "string"
    }],
    "StatelessCustomActions": [{
      "ActionDefinition": {
        "PublishMetricAction": {
          "Dimensions": [{
            "Value": "string"
          }]
        }
      }
    ]},
    "ActionName": "string"
  }],
  "StatelessDefaultActions": ["string"],
  "StatelessFragmentDefaultActions": ["string"],
  "StatelessRuleGroupReferences": [{
    "Priority": "number",
    "ResourceArn": "string"
  }]
},
"FirewallPolicyArn": "string",
"FirewallPolicyId": "string",
"FirewallPolicyName": "string"
},
"AwsNetworkFirewallRuleGroup (p. 249)": {
  "Capacity": "number",
  "Description": "string",
  "RuleGroup": {
    "RulesSource": {
      "RulesSourceList": {
        "GeneratedRulesType": "string",
        "Targets": ["string"],
        "TargetTypes": ["string"]
      }
    },
    "RulesString": "string",
    "StatefulRules": [{
      "Action": "string",
      "Header": {
        "Destination": "string",
        "DestinationPort": "string",
        "Direction": "string",
        "Protocol": "string",
        "Source": "string",
        "SourcePort": "string"
      }
    }],
    "RuleOptions": [{
      "Keyword": "string",
      "Settings": ["string"]
    }]
  }],
  "StatelessRulesAndCustomActions": {
    "CustomActions": [{
      "ActionDefinition": {
        "PublishMetricAction": {
          "Dimensions": [{
            "Value": "string"
          }]
        }
      }
    ]}
  }
}

```

```

    }}
  }
},
"ActionName": "string"
}],
"StatelessRules": [{
  "Priority": "number",
  "RuleDefinition": {
    "Actions": ["string"],
    "MatchAttributes": {
      "DestinationPorts": [{
        "FromPort": "number",
        "ToPort": "number"
      }],
      "Destinations": [{
        "AddressDefinition": "string"
      }],
      "Protocols": ["number"],
      "SourcePorts": [{
        "FromPort": "number",
        "ToPort": "number"
      }],
      "Sources": [{
        "AddressDefinition": "string"
      }],
      "TcpFlags": [{
        "Flags": ["string"],
        "Masks": ["string"]
      }]
    }
  }
}],
"RuleVariables": {
  "IpSets": {
    "Definition": ["string"]
  },
  "PortSets": {
    "Definition": ["string"]
  }
},
"RuleGroupArn": "string",
"RuleGroupId": "string",
"RuleGroupName": "string",
"Type": "string"
},
"AwsOpenSearchServiceDomain (p. 251)": {
  "AccessPolicies": "string",
  "AdvancedSecurityOptions": {
    "Enabled": "boolean",
    "InternalUserDatabaseEnabled": "boolean",
    "MasterUserOptions": {
      "MasterUserArn": "string",
      "MasterUserName": "string",
      "MasterUserPassword": "string"
    }
  },
  "Arn": "string",
  "ClusterConfig": {
    "DedicatedMasterCount": "number",
    "DedicatedMasterEnabled": "boolean",
    "DedicatedMasterType": "string",
    "InstanceCount": "number",
    "InstanceType": "string",

```

```

    "WarmCount": "number",
    "WarmEnabled": "boolean",
    "WarmType": "string",
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": "number"
    },
    "ZoneAwarenessEnabled": "boolean"
  },
  "DomainEndpoint": "string",
  "DomainEndpointOptions": {
    "CustomEndpoint": "string",
    "CustomEndpointCertificateArn": "string",
    "CustomEndpointEnabled": "boolean",
    "EnforceHTTPS": "boolean",
    "TLSSecurityPolicy": "string"
  },
  "DomainEndpoints": {
    "string": "string"
  },
  "DomainName": "string",
  "EncryptionAtRestOptions": {
    "Enabled": "boolean",
    "KmsKeyId": "string"
  },
  "EngineVersion": "string",
  "Id": "string",
  "LogPublishingOptions": {
    "AuditLogs": {
      "CloudWatchLogsLogGroupArn": "string",
      "Enabled": "boolean"
    },
    "IndexSlowLogs": {
      "CloudWatchLogsLogGroupArn": "string",
      "Enabled": "boolean"
    },
    "SearchSlowLogs": {
      "CloudWatchLogsLogGroupArn": "string",
      "Enabled": "boolean"
    }
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": "boolean"
  },
  "ServiceSoftwareOptions": {
    "AutomatedUpdateDate": "string",
    "Cancellable": "boolean",
    "CurrentVersion": "string",
    "Description": "string",
    "NewVersion": "string",
    "OptionalDeployment": "boolean",
    "UpdateAvailable": "boolean",
    "UpdateStatus": "string"
  },
  "VpcOptions": {
    "SecurityGroupIds": ["string"],
    "SubnetIds": ["string"]
  }
},
"AwsRdsDbCluster (p. 252)": {
  "ActivityStreamStatus": "string",
  "AllocatedStorage": "number",
  "AssociatedRoles": [{
    "RoleArn": "string",
    "Status": "string"
  }],
  "AvailabilityZones": ["string"],

```

```

"BackupRetentionPeriod": "integer",
"ClusterCreateTime": "string",
"CopyTagsToSnapshot": "boolean",
"CrossAccountClone": "boolean",
"CustomEndpoints": ["string"],
"DatabaseName": "string",
"DbClusterIdentifier": "string",
"DbClusterMembers": [{
  "DbClusterParameterGroupStatus": "string",
  "DbInstanceIdentifier": "string",
  "IsClusterWriter": "boolean",
  "PromotionTier": "integer"
}],
"DbClusterOptionGroupMemberships": [{
  "DbClusterOptionGroupName": "string",
  "Status": "string"
}],
"DbClusterParameterGroup": "string",
"DbClusterResourceId": "string",
"DbSubnetGroup": "string",
"DeletionProtection": "boolean",
"DomainMemberships": [{
  "Domain": "string",
  "Fqdn": "string",
  "IamRoleName": "string",
  "Status": "string"
}],
"EnabledCloudwatchLogsExports": ["string"],
"Endpoint": "string",
"Engine": "string",
"EngineMode": "string",
"EngineVersion": "string",
"HostedZoneId": "string",
"HttpEndpointEnabled": "boolean",
"IamDatabaseAuthenticationEnabled": "boolean",
"KmsKeyId": "string",
"MasterUsername": "string",
"MultiAz": "boolean",
"Port": "integer",
"PreferredBackupWindow": "string",
"PreferredMaintenanceWindow": "string",
"ReaderEndpoint": "string",
"ReadReplicaIdentifiers": ["string"],
"Status": "string",
"StorageEncrypted": "boolean",
"VpcSecurityGroups": [{
  "Status": "string",
  "VpcSecurityGroupId": "string"
}]
}],
"AwsRdsDbClusterSnapshot (p. 254)": {
  "AllocatedStorage": "integer",
  "AvailabilityZones": ["string"],
  "ClusterCreateTime": "string",
  "DbClusterIdentifier": "string",
  "DbClusterSnapshotIdentifier": "string",
  "Engine": "string",
  "EngineVersion": "string",
  "IamDatabaseAuthenticationEnabled": "boolean",
  "KmsKeyId": "string",
  "LicenseModel": "string",
  "MasterUsername": "string",
  "PercentProgress": "integer",
  "Port": "integer",
  "SnapshotCreateTime": "string",
  "SnapshotType": "string",

```

```

    "Status": "string",
    "StorageEncrypted": "boolean",
    "VpcId": "string"
  },
  "AwsRdsDbInstance (p. 254)": {
    "AllocatedStorage": "number",
    "AssociatedRoles": [{
      "RoleArn": "string",
      "FeatureName": "string",
      "Status": "string"
    }],
    "AutoMinorVersionUpgrade": "boolean",
    "AvailabilityZone": "string",
    "BackupRetentionPeriod": "number",
    "CACertificateIdentifier": "string",
    "CharacterSetName": "string",
    "CopyTagsToSnapshot": "boolean",
    "DBClusterIdentifier": "string",
    "DBInstanceClass": "string",
    "DBInstanceIdentifier": "string",
    "DbInstancePort": "number",
    "DbInstanceStatus": "string",
    "DbiResourceId": "string",
    "DBName": "string",
    "DbParameterGroups": [{
      "DbParameterGroupName": "string",
      "ParameterApplyStatus": "string"
    }],
    "DbSecurityGroups": ["string"],
    "DbSubnetGroup": {
      "DbSubnetGroupArn": "string",
      "DbSubnetGroupDescription": "string",
      "DbSubnetGroupName": "string",
      "SubnetGroupStatus": "string",
      "Subnets": [{
        "SubnetAvailabilityZone": {
          "Name": "string"
        },
        "SubnetIdentifier": "string",
        "SubnetStatus": "string"
      }],
      "VpcId": "string"
    },
    "DeletionProtection": "boolean",
    "Endpoint": {
      "Address": "string",
      "Port": "number",
      "HostedZoneId": "string"
    },
    "DomainMemberships": [{
      "Domain": "string",
      "Fqdn": "string",
      "IamRoleName": "string",
      "Status": "string"
    }],
    "EnabledCloudwatchLogsExports": ["string"],
    "Engine": "string",
    "EngineVersion": "string",
    "EnhancedMonitoringResourceArn": "string",
    "IAMDatabaseAuthenticationEnabled": "boolean",
    "InstanceCreateTime": "string",
    "Iops": "number",
    "KmsKeyId": "string",
    "LatestRestorableTime": "string",
    "LicenseModel": "string",
    "ListenerEndpoint": {

```

```

    "Address": "string",
    "HostedZoneId": "string",
    "Port": "number"
  },
  "MasterUsername": "admin",
  "MaxAllocatedStorage": "number",
  "MonitoringInterval": "number",
  "MonitoringRoleArn": "string",
  "MultiAz": "boolean",
  "OptionGroupMemberships": [{
    "OptionGroupName": "string",
    "Status": "string"
  }],
  "PendingModifiedValues": {
    "AllocatedStorage": "number",
    "BackupRetentionPeriod": "number",
    "CaCertificateIdentifier": "string",
    "DbInstanceClass": "string",
    "DbInstanceIdentifier": "string",
    "DbSubnetGroupName": "string",
    "EngineVersion": "string",
    "Iops": "number",
    "LicenseModel": "string",
    "MasterUserPassword": "string",
    "MultiAZ": "boolean",
    "PendingCloudWatchLogsExports": {
      "LogTypesToDisable": ["string"],
      "LogTypesToEnable": ["string"]
    },
    "Port": "number",
    "ProcessorFeatures": [{
      "Name": "string",
      "Value": "string"
    }],
    "StorageType": "string"
  },
  "PerformanceInsightsEnabled": "boolean",
  "PerformanceInsightsKmsKeyId": "string",
  "PerformanceInsightsRetentionPeriod": "number",
  "PreferredBackupWindow": "string",
  "PreferredMaintenanceWindow": "string",
  "ProcessorFeatures": [{
    "Name": "string",
    "Value": "string"
  }],
  "PromotionTier": "number",
  "PubliclyAccessible": "boolean",
  "ReadReplicaDBClusterIdentifiers": ["string"],
  "ReadReplicaDBInstanceIdentifiers": ["string"],
  "ReadReplicaSourceDBInstanceIdentifier": "string",
  "SecondaryAvailabilityZone": "string",
  "StatusInfos": [{
    "Message": "string",
    "Normal": "boolean",
    "Status": "string",
    "StatusType": "string"
  }],
  "StorageEncrypted": "boolean",
  "TdeCredentialArn": "string",
  "Timezone": "string",
  "VpcSecurityGroups": [{
    "VpcSecurityGroupId": "string",
    "Status": "string"
  }],
  "AwsRdsDbSnapshot (p. 257)": {

```

```

"AllocatedStorage": "integer",
"AvailabilityZone": "string",
"DbInstanceIdentifier": "string",
"DbiResourceId": "string",
"DbSnapshotIdentifier": "string",
"Encrypted": "boolean",
"Engine": "string",
"EngineVersion": "string",
"IamDatabaseAuthenticationEnabled": "boolean",
"InstanceCreateTime": "string",
"Iops": "number",
"KmsKeyId": "string",
"LicenseModel": "string",
"MasterUsername": "string",
"OptionGroupName": "string",
"PercentProgress": "integer",
"Port": "integer",
"ProcessorFeatures": [],
"SnapshotCreateTime": "string",
"SnapshotType": "string",
"SourceDbSnapshotIdentifier": "string",
"SourceRegion": "string",
>Status": "string",
"StorageType": "string",
"TdeCredentialArn": "string",
"Timezone": "string",
"VpcId": "string"
},
"AwsRdsEventSubscription (p. 258)": {
  "CustomerAwsId": "string",
  "CustSubscriptionId": "string",
  "Enabled": "boolean",
  "EventCategoriesList": ["string"],
  "EventSubscriptionArn": "string",
  "SnsTopicArn": "string",
  "SourceIdsList": ["string"],
  "SourceType": "string",
  "Status": "string",
  "SubscriptionCreationTime": "string"
},
"AwsRedshiftCluster (p. 258)": {
  "AllowVersionUpgrade": "boolean",
  "AutomatedSnapshotRetentionPeriod": "number",
  "AvailabilityZone": "string",
  "ClusterAvailabilityStatus": "string",
  "ClusterCreateTime": "string",
  "ClusterIdentifier": "string",
  "ClusterNodes": [{
    "NodeRole": "string",
    "PrivateIpAddress": "string",
    "PublicIpAddress": "string"
  }],
  "ClusterParameterGroups": [{
    "ClusterParameterStatusList": [{
      "ParameterApplyErrorDescription": "string",
      "ParameterApplyStatus": "string",
      "ParameterName": "string"
    }],
    "ParameterApplyStatus": "string",
    "ParameterGroupName": "string"
  }],
  "ClusterPublicKey": "string",
  "ClusterRevisionNumber": "string",
  "ClusterSecurityGroups": [{
    "ClusterSecurityGroupName": "string",
    "Status": "string"
  }],

```



```

    }],
    "ClusterSnapshotCopyStatus": {
      "DestinationRegion": "string",
      "ManualSnapshotRetentionPeriod": "number",
      "RetentionPeriod": "number",
      "SnapshotCopyGrantName": "string"
    },
    "ClusterStatus": "string",
    "ClusterSubnetGroupName": "string",
    "ClusterVersion": "string",
    "DBName": "string",
    "DeferredMaintenanceWindows": [{
      "DeferMaintenanceEndTime": "string",
      "DeferMaintenanceIdentifier": "string",
      "DeferMaintenanceStartTime": "string"
    }],
    "ElasticIpStatus": {
      "ElasticIp": "string",
      "Status": "string"
    },
    "ElasticResizeNumberOfNodeOptions": "string",
    "Encrypted": "boolean",
    "Endpoint": {
      "Address": "string",
      "Port": "number"
    },
    "EnhancedVpcRouting": "boolean",
    "ExpectedNextSnapshotScheduleTime": "string",
    "ExpectedNextSnapshotScheduleTimeStatus": "string",
    "HsmStatus": {
      "HsmClientCertificateIdentifier": "string",
      "HsmConfigurationIdentifier": "string",
      "Status": "string"
    },
    "IamRoles": [{
      "ApplyStatus": "string",
      "IamRoleArn": "string"
    }],
    "KmsKeyId": "string",
    "LoggingStatus": {
      "BucketName": "string",
      "LastFailureMessage": "string",
      "LastFailureTime": "string",
      "LastSuccessfulDeliveryTime": "string",
      "LoggingEnabled": "boolean",
      "S3KeyPrefix": "string"
    },
    "MaintenanceTrackName": "string",
    "ManualSnapshotRetentionPeriod": "number",
    "MasterUsername": "string",
    "NextMaintenanceWindowStartTime": "string",
    "NodeType": "string",
    "NumberOfNodes": "number",
    "PendingActions": ["string"],
    "PendingModifiedValues": {
      "AutomatedSnapshotRetentionPeriod": "number",
      "ClusterIdentifier": "string",
      "ClusterType": "string",
      "ClusterVersion": "string",
      "EncryptionType": "string",
      "EnhancedVpcRouting": "boolean",
      "MaintenanceTrackName": "string",
      "MasterUserPassword": "string",
      "NodeType": "string",
      "NumberOfNodes": "number",
      "PubliclyAccessible": "string"
    }
  }
}

```

```

},
"PreferredMaintenanceWindow": "string",
"PubliclyAccessible": "boolean",
"ResizeInfo": {
  "AllowCancelResize": "boolean",
  "ResizeType": "string"
},
},
"RestoreStatus": {
  "CurrentRestoreRateInMegaBytesPerSecond": "number",
  "ElapsedTimeInSeconds": "number",
  "EstimatedTimeToCompletionInSeconds": "number",
  "ProgressInMegaBytes": "number",
  "SnapshotSizeInMegaBytes": "number",
  "Status": "string"
},
},
"SnapshotScheduleIdentifier": "string",
"SnapshotScheduleState": "string",
"VpcId": "string",
"VpcSecurityGroups": [{
  "Status": "string",
  "VpcSecurityGroupId": "string"
}]
}],
},
"AwsS3AccountPublicAccessBlock (p. 261)": {
  "BlockPublicAcls": "boolean",
  "BlockPublicPolicy": "boolean",
  "IgnorePublicAcls": "boolean",
  "RestrictPublicBuckets": "boolean"
},
},
"AwsS3Bucket (p. 262)": {
  "AccessControlList": "string",
  "BucketLifecycleConfiguration": {
    "Rules": [{
      "AbortIncompleteMultipartUpload": {
        "DaysAfterInitiation": "number"
      },
      "ExpirationDate": "string",
      "ExpirationInDays": "number",
      "ExpiredObjectDeleteMarker": "boolean",
      "Filter": {
        "Predicate": {
          "Operands": [{
            "Prefix": "string",
            "Type": "string"
          },
          {
            "Tag": {
              "Key": "string",
              "Value": "string"
            },
            "Type": "string"
          }
        ]
      },
      "Type": "string"
    }
  ],
  "Type": "string"
},
},
},
"Id": "string",
"NoncurrentVersionExpirationInDays": "number",
"NoncurrentVersionTransitions": [{
  "Days": "number",
  "StorageClass": "string"
}],
},
"Prefix": "string",
"Status": "string",
"Transitions": [{
  "Date": "string",

```

```

    "Days": "number",
    "StorageClass": "string"
  }]
}]
},
"BucketLoggingConfiguration": {
  "DestinationBucketName": "string",
  "LogFilePrefix": "string"
},
"BucketNotificationConfiguration": {
  "Configurations": [{
    "Destination": "string",
    "Events": ["string"],
    "Filter": {
      "S3KeyFilter": {
        "FilterRules": [{
          "Name": "string",
          "Value": "string"
        }]
      }
    }
  ],
  "Type": "string"
}]
},
"BucketVersioningConfiguration": {
  "IsMfaDeleteEnabled": "boolean",
  "Status": "string"
},
"BucketWebsiteConfiguration": {
  "ErrorDocument": "string",
  "IndexDocumentSuffix": "string",
  "RedirectAllRequestsTo": {
    "HostName": "string",
    "Protocol": "string"
  }
},
"RoutingRules": [{
  "Condition": {
    "HttpErrorCodeReturnedEquals": "string",
    "KeyPrefixEquals": "string"
  },
  "Redirect": {
    "HostName": "string",
    "HttpRedirectCode": "string",
    "Protocol": "string",
    "ReplaceKeyPrefixWith": "string",
    "ReplaceKeyWith": "string"
  }
}]
},
"CreatedAt": "string",
"OwnerAccountId": "string",
"OwnerId": "string",
"OwnerName": "string",
"PublicAccessBlockConfiguration": {
  "BlockPublicAcls": "boolean",
  "BlockPublicPolicy": "boolean",
  "IgnorePublicAcls": "boolean",
  "RestrictPublicBuckets": "boolean"
},
"ServerSideEncryptionConfiguration": {
  "Rules": [{
    "ApplyServerSideEncryptionByDefault": {
      "KMSEncryption": "string",
      "SSEAlgorithm": "string"
    }
  }]
}]

```

```

    }
  },
  "AwsS3Object (p. 263)": {
    "ContentType": "string",
    "ETag": "string",
    "LastModified": "string",
    "ServerSideEncryption": "string",
    "SSEKMSKeyId": "string",
    "VersionId": "string"
  },
  "AwsSecretsManagerSecret (p. 264)": {
    "Deleted": "boolean",
    "Description": "string",
    "KmsKeyId": "string",
    "Name": "string",
    "RotationEnabled": "boolean",
    "RotationLambdaArn": "string",
    "RotationOccurredWithinFrequency": "boolean",
    "RotationRules": {
      "AutomaticallyAfterDays": "integer"
    }
  },
  "AwsSnsTopic (p. 264)": {
    "ApplicationSuccessFeedbackRoleArn": "string",
    "FirehoseFailureFeedbackRoleArn": "string",
    "FirehoseSuccessFeedbackRoleArn": "string",
    "HttpFailureFeedbackRoleArn": "string",
    "HttpSuccessFeedbackRoleArn": "string",
    "KmsMasterKeyId": "string",
    "Owner": "string",
    "SqsFailureFeedbackRoleArn": "string",
    "SqsSuccessFeedbackRoleArn": "string",
    "Subscription": {
      "Endpoint": "string",
      "Protocol": "string"
    },
    "TopicName": "string"
  },
  "AwsSqsQueue (p. 265)": {
    "DeadLetterTargetArn": "string",
    "KmsDataKeyReusePeriodSeconds": "number",
    "KmsMasterKeyId": "string",
    "QueueName": "string"
  },
  "AwsSsmPatchCompliance (p. 265)": {
    "Patch": {
      "ComplianceSummary": {
        "ComplianceType": "string",
        "CompliantCriticalCount": "integer",
        "CompliantHighCount": "integer",
        "CompliantInformationalCount": "integer",
        "CompliantLowCount": "integer",
        "CompliantMediumCount": "integer",
        "CompliantUnspecifiedCount": "integer",
        "ExecutionType": "string",
        "NonCompliantCriticalCount": "integer",
        "NonCompliantHighCount": "integer",
        "NonCompliantInformationalCount": "integer",
        "NonCompliantLowCount": "integer",
        "NonCompliantMediumCount": "integer",
        "NonCompliantUnspecifiedCount": "integer",
        "OverallSeverity": "string",
        "PatchBaselineId": "string",
        "PatchGroup": "string",
        "Status": "string"
      }
    }
  }
}

```

```

    }
  },
  "AwsWafRateBasedRule (p. 266)": {
    "MatchPredicates": [{
      "DataId": "string",
      "Negated": "boolean",
      "Type": "string"
    }],
    "MetricName": "string",
    "Name": "string",
    "RateKey": "string",
    "RateLimit": "number",
    "RuleId": "string"
  },
  "AwsWafRegionalRateBasedRule (p. 266)": {
    "MatchPredicates": [{
      "DataId": "string",
      "Negated": "boolean",
      "Type": "string"
    }],
    "MetricName": "string",
    "Name": "string",
    "RateKey": "string",
    "RateLimit": "number",
    "RuleId": "string"
  },
  "AwsWafRegionalRule (p. 267)": {
    "MetricName": "string",
    "Name": "string",
    "RuleId": "string",
    "PredicateList": [{
      "DataId": "string",
      "Negated": "boolean",
      "Type": "string"
    }]
  },
  "AwsWafRegionalRuleGroup (p. 267)": {
    "MetricName": "string",
    "Name": "string",
    "RuleGroupId": "string",
    "Rules": [{
      "Action": {
        "Type": "string"
      },
      "Priority": "number",
      "RuleId": "string",
      "Type": "string"
    }]
  },
  "AwsWafRegionalWebAcl (p. 268)": {
    "DefaultAction": "string",
    "MetricName": "string",
    "Name": "string",
    "RulesList": [{
      "Action": {
        "Type": "string"
      },
      "Priority": "number",
      "RuleId": "string",
      "Type": "string",
      "ExcludedRules": [{
        "ExclusionType": "string",
        "RuleId": "string"
      }]
    }],
    "OverrideAction": {
      "Type": "string"
    }
  }
}

```

```

    }
  }],
  "WebAclId": "string"
},
"AwsWafRule (p. 268)": {
  "MetricName": "string",
  "Name": "string",
  "PredicateList": [{
    "DataId": "string",
    "Negated": "boolean",
    "Type": "string"
  }],
  "RuleId": "string"
},
"AwsWafRuleGroup (p. 269)": {
  "MetricName": "string",
  "Name": "string",
  "RuleGroupId": "string",
  "Rules": [{
    "Action": {
      "Type": "string"
    },
    "Priority": "number",
    "RuleId": "string",
    "Type": "string"
  }],
},
"AwsWafWebAcl (p. 269)": {
  "DefaultAction": "string",
  "Name": "string",
  "Rules": [{
    "Action": {
      "Type": "string"
    },
    "ExcludedRules": [{
      "RuleId": "string"
    }],
    "OverrideAction": {
      "Type": "string"
    },
    "Priority": "number",
    "RuleId": "string",
    "Type": "string"
  }],
  "WebAclId": "string"
},
"AwsXrayEncryptionConfig (p. 270)": {
  "KeyId": "string",
  "Status": "string",
  "Type": "string"
},
"Container (p. 270)": {
  "ContainerRuntime": "string",
  "ImageId": "string",
  "ImageName": "string",
  "LaunchedAt": "string",
  "Name": "string",
  "Privileged": "boolean",
  "VolumeMounts": [{
    "Name": "string",
    "MountPath": "string"
  }],
},
"Other (p. 270)": {
  "string": "string"
},

```

```

    "Id": "string",
    "Partition": "string",
    "Region": "string",
    "ResourceRole": "string",
    "Tags": {
      "string": "string"
    },
    "Type": "string"
  }],
  "SchemaVersion": "string",
  "Severity (p. 271)": {
    "Label": "string",
    "Normalized": "number",
    "Original": "string",
    "Product": "number"
  },
  "Sample": "boolean",
  "SourceUrl": "string",
  "Threats (p. 273)": [{
    "FilePaths": [{
      "FileName": "string",
      "FilePath": "string",
      "Hash": "string",
      "ResourceId": "string",
    }],
    "ItemCount": "number",
    "Name": "string",
    "Severity": "string"
  }],
  "ThreatIntelIndicators (p. 273)": [{
    "Category": "string",
    "LastObservedAt": "string",
    "Source": "string",
    "SourceUrl": "string",
    "Type": "string",
    "Value": "string"
  }],
  "Title": "string",
  "Types": ["string"],
  "UpdatedAt": "string",
  "UserDefinedFields": {
    "string": "string"
  },
  "VerificationState": "string",
  "Vulnerabilities (p. 273)": [{
    "Cvss": [{
      "Adjustments": [{
        "Metric": "string",
        "Reason": "string"
      }],
      "BaseScore": "number",
      "BaseVector": "string",
      "Source": "string",
      "Version": "string"
    }],
    "Id": "string",
    "ReferenceUrls": ["string"],
    "RelatedVulnerabilities": ["string"],
    "Vendor": {
      "Name": "string",
      "Url": "string",
      "VendorCreatedAt": "string",
      "VendorSeverity": "string",
      "VendorUpdatedAt": "string"
    },
    "VulnerablePackages": [{

```

```
    "Architecture": "string",
    "Epoch": "string",
    "FilePath": "string",
    "Name": "string",
    "PackageManager": "string",
    "Release": "string",
    "Version": "string"
  }]
},
"Workflow (p. 274)": {
  "Status": "string"
},
"WorkflowState": "string"
}
]
```

ASFF examples

The following sections contain details and examples of objects in the AWS Security Finding Format. Choose an object to see more information about it.

Topics

- [Required attributes \(p. 176\)](#)
- [Other top-level attributes \(p. 180\)](#)
- [Action \(p. 188\)](#)
- [Compliance \(p. 189\)](#)
- [FindingProviderFields \(p. 189\)](#)
- [Malware \(p. 189\)](#)
- [Network \(Deprecated\) \(p. 190\)](#)
- [NetworkPath \(p. 190\)](#)
- [Note \(p. 191\)](#)
- [PatchSummary \(p. 191\)](#)
- [Process \(p. 192\)](#)
- [RelatedFindings \(p. 192\)](#)
- [Remediation \(p. 192\)](#)
- [Resources \(p. 193\)](#)
- [Severity \(p. 271\)](#)
- [Threats \(p. 273\)](#)
- [ThreatIntelIndicators \(p. 273\)](#)
- [Vulnerabilities \(p. 273\)](#)
- [Workflow \(p. 274\)](#)

Required attributes

The following attributes are required for all findings in Security Hub. For more information about these required attributes, see [AwsSecurityFinding](#) in the *AWS Security Hub API Reference*.

AwsAccountId

The AWS account ID that the finding applies to.

Example


```
"AwsAccountId": "111111111111"
```

CreatedAt

Indicates when the potential security issue captured by a finding was created.

Example

```
"CreatedAt": "2017-03-22T13:22:13.933Z"
```

Note

Findings are deleted 90 days after the most recent update or 90 days after the creation date if no update occurs. To store findings for longer than 90 days, you can configure a rule in Amazon EventBridge that routes findings to your S3 bucket.

Description

A finding's description. This field can be nonspecific boilerplate text or details that are specific to the instance of the finding.

Example

```
"Description": "The version of openssl found on instance i-abcd1234 is known to contain a vulnerability."
```

GeneratorId

The identifier for the solution-specific component (a discrete unit of logic) that generated a finding. In various solutions from security findings products, this generator can be called a rule, a check, a detector, a plugin, and so on.

Example

```
"GeneratorId": "acme-vuln-9ab348"
```

Id

The product-specific identifier for a finding.

Example

```
"Id": "us-west-2/111111111111/98aebb2207407c87f51e89943f12b1ef"
```

ProductArn

The ARN generated by Security Hub that uniquely identifies a third-party findings product after the product is registered with Security Hub.

The format of this field is `arn:partition:securityhub:region:account-id:product/company-id/product-id`.

- For AWS services that are integrated with Security Hub, the `company-id` must be "aws", and the `product-id` must be the AWS public service name. Because AWS products and services aren't associated with an account, the `account-id` section of the ARN is empty. AWS services that are not yet integrated with Security Hub are considered third-party products.
- For public products, the `company-id` and `product-id` must be the ID values specified at the time of registration.

- For private products, the `company-id` must be the account ID. The `product-id` must be the reserved word "default" or the ID that was specified at the time of registration.

Example:

```
// Private ARN
  "ProductArn": "arn:aws:securityhub:us-east-1:111111111111:product/111111111111/default"

// Public ARN
  "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty"
  "ProductArn": "arn:aws:securityhub:us-west-2:222222222222:product/generico/secure-pro"
```

Resources (p. 193)

A set of resource data types that describe the resources that the finding refers to.

Example

```
"Resources": [
  {
    "Type": "AwsEc2Instance",
    "Id": "i-cafebabe",
    "Partition": "aws",
    "Region": "us-west-2",
    "Tags": {
      "billingCode": "Lotus-1-2-3",
      "needsPatching": "true"
    },
    "Details": {
      "IamInstanceProfileArn": "arn:aws:iam::123456789012:role/IamInstanceProfileArn",
      "ImageId": "ami-79fd7eee",
      "IPv4Addresses": ["1.1.1.1"],
      "IPv6Addresses": ["2001:db8:1234:1a2b::123"],
      "KeyName": "testkey",
      "LaunchedAt": "2018-09-29T01:25:54Z",
      "MetadataOptions": {
        "HttpEndpoint": "enabled",
        "HttpProtocolIpv6": "enabled",
        "HttpPutResponseHopLimit": 1,
        "HttpTokens": "optional",
        "InstanceMetadataTags": "disabled"
      },
      "NetworkInterfaces": [
        {
          "NetworkInterfaceId": "eni-e5aa89a3"
        }
      ],
      "SubnetId": "PublicSubnet",
      "Type": "i3.xlarge",
      "VirtualizationType": "hvm",
      "VpcId": "TestVPCIPv6"
    }
  }
]
```

SchemaVersion

The schema version that a finding is formatted for. The value of this field must be one of the officially published versions identified by AWS.

In the current release, the AWS Security Finding Format schema version is 2018-10-08.

Example

```
"SchemaVersion": "2018-10-08"
```

Severity (p. 271)

A finding's severity.

The finding must have either `Label` or `Normalized` populated. `Label` is the preferred attribute. `Normalized` is no longer relevant. Security Hub populates the `Normalized` attribute but does not otherwise use it. If neither attribute is populated, then the finding is invalid.

Severity should only be updated by [BatchUpdateFindings](#).

To provide severity information, finding providers should use the `Severity` object under [FindingProviderFields](#) (p. 189). See the section called "Using [FindingProviderFields](#)" (p. 120).

Example

```
"Severity": {  
  "Label": "CRITICAL",  
  "Original": "8.3"  
}
```

Title

A finding's title. This field can contain nonspecific boilerplate text or details that are specific to this instance of the finding.

Types

One or more finding types in the format of `namespace/category/classifier` that classify a finding.

Types should only be updated using [BatchUpdateFindings](#).

Finding providers who want to provide a value for `Types` should use the `Types` attribute under [FindingProviderFields](#) (p. 189). See the section called "Using [FindingProviderFields](#)" (p. 120).

- `namespace` must be a value from the predefined set of namespace values.

Valid values: Software and Configuration Checks | TTPs | Effects | Unusual Behaviors | Sensitive Data Identifications

- `category` might be any value, but it is recommended that finding products use categories from the finding type taxonomy in [Types taxonomy for ASFF](#) (p. 274).
- `classifier` might be any value, but it is recommended that finding providers use the identifier exactly as defined by published standards whenever possible.

Namespaces are required for all finding types, but categories and classifiers are optional. If you specify a classifier, you must also specify a category.

The '/' character is reserved and must not be used in a category or classifier. Escaping the '/' character is not supported.

Example

```
"Types": [  
  "
```

```
    "Software and Configuration Checks/Vulnerabilities/CVE"  
  ]
```

UpdatedAt

Indicates when the finding provider last updated the finding record.

This timestamp reflects the time when the finding record was last or most recently updated. Consequently, it can differ from the `LastObservedAt` timestamp, which reflects when the event or vulnerability was last or most recently observed.

When you update the finding record, you must update this timestamp to the current timestamp. Upon creation of a finding record, the `CreatedAt` and `UpdatedAt` timestamps must be the same. After an update to the finding record, the value of this field must be greater than all of the previous values that it contained.

Note that `UpdatedAt` is not updated by changes from [BatchUpdateFindings](#). It is only updated by [BatchImportFindings](#).

Findings are deleted 90 days after the most recent update or 90 days after the creation date if no update occurs. To store findings for longer than 90 days, you can configure a rule in CloudWatch Events that routes findings to your Amazon S3 bucket.

Other top-level attributes

These are top-level attributes that aren't required by the AWS Security Finding Format. For more information about these attributes, see [AwsSecurityFinding](#) in the *AWS Security Hub API Reference*.

Action (p. 188)

Provides details about an action that affects or that was taken on a resource.

CompanyName

The name of the company for the product that generated the finding. For control-based findings, the company is AWS.

Security Hub populates this attribute automatically for each finding. You cannot update it using [BatchImportFindings](#) or [BatchUpdateFindings](#). The exception to this is when you use a custom integration. See [the section called "Using custom product integrations" \(p. 321\)](#).

When you use the Security Hub console to filter findings by company name, you use this attribute.

When you use the Security Hub API to filter findings by company name, you use the `aws/securityhub/CompanyName` attribute under `ProductFields`.

Security Hub does not synchronize those two attributes.

Compliance (p. 189)

Finding details related to a control. Only returned for findings generated from a control.

Example

```
"Compliance": {  
  "RelatedRequirements": ["Req1", "Req2"],  
  "Status": "PASSED",  
  "StatusReasons": [  
    {  
      "ReasonCode": "CLOUDWATCH_ALARMS_NOT_PRESENT";
```

```
        "Description": "CloudWatch alarms do not exist in the account"  
      }  
    ]  
  }
```

Confidence

The likelihood that a finding accurately identifies the behavior or issue that it was intended to identify.

Confidence should only be updated using [BatchUpdateFindings](#).

Finding providers who want to provide a value for Confidence should use the Confidence attribute under [FindingProviderFields](#) (p. 189). See [the section called "Using FindingProviderFields"](#) (p. 120).

Confidence is scored on a 0–100 basis using a ratio scale, where 0 means 0-percent confidence and 100 means 100-percent confidence.

However, a data exfiltration detection based on a statistical deviation of network traffic has a much lower confidence because an actual exfiltration hasn't been verified.

Example

```
"Confidence": 42
```

Criticality

The level of importance that is assigned to the resources that are associated with the finding. A score of 0 means that the underlying resources have no criticality, and a score of 100 is reserved for the most critical resources.

Criticality should only be updated by [BatchUpdateFindings](#). It should not be updated by [BatchImportFindings](#).

Finding providers who want to provide a value for Criticality should use the Criticality attribute under [FindingProviderFields](#) (p. 189). See [the section called "Using FindingProviderFields"](#) (p. 120).

Criticality is scored on a 0–100 basis, using a ratio scale that supports only full integers. A score of 0 means that the underlying resources have no criticality, and a score of 100 is reserved for the most critical resources.

At a high level, when assessing criticality, you need to consider the following:

- Which findings impact resources that are more critical than other resources?
- How much more critical are those resources compared to other resources?

For each resource, consider the following:

- Does the affected resource contain sensitive data (for example, an S3 bucket with PII)?
- Does the affected resource enable an adversary to deepen their access or extend their capabilities to carry out additional malicious activity (for example, a compromised sysadmin account)?
- Is the resource a business-critical asset (for example, a key business system that if compromised could have significant revenue impact)?

You can use the following guidelines:

- A resource powering mission-critical systems or containing highly sensitive data can be scored in the 75–100 range.

- A resource powering important (but not critical systems) or containing moderately important data can be scored in the 25–75 range.
- A resource powering unimportant systems or containing nonsensitive data should be scored in the 0–24 range.

Example

```
"Criticality": 99
```

FindingProviderFields (p. 189)

In [BatchImportFindings](#) requests, finding providers use `FindingProviderFields` to provide values for attributes that should only be updated by [BatchUpdateFindings](#). `FindingProviderFields` includes the following attributes:

- Confidence
- Criticality
- RelatedFindings
- Severity
- Types

`FindingProviderFields` can only be updated by [BatchImportFindings](#). It cannot be updated by [BatchUpdateFindings](#).

For details on how Security Hub handles updates from [BatchImportFindings](#) to `FindingProviderFields` and to the corresponding top-level attributes, see [the section called "Using FindingProviderFields" \(p. 120\)](#).

Example

```
"FindingProviderFields": {
  "Confidence": 42,
  "Criticality": 99,
  "RelatedFindings": [
    {
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
      "Id": "123e4567-e89b-12d3-a456-426655440000"
    }
  ],
  "Severity": {
    "Label": "MEDIUM",
    "Original": "MEDIUM"
  },
  "Types": [ "Software and Configuration Checks/Vulnerabilities/CVE" ]
}
```

FirstObservedAt

Indicates when the potential security issue captured by a finding was first observed.

This timestamp reflects the time of when the event or vulnerability was first observed. Consequently, it can differ from the `CreatedAt` timestamp, which reflects the time this finding record was created.

This timestamp should be immutable between updates of the finding record but can be updated if a more accurate timestamp is determined.

Example

```
"FirstObservedAt": "2017-03-22T13:22:13.933Z"
```

LastObservedAt

Indicates when the potential security issue that was captured by a finding was most recently observed by the security findings product.

This timestamp reflects the time when the event or vulnerability was last or most recently observed. Consequently, it can differ from the `UpdatedAt` timestamp, which reflects when this finding record was last or most recently updated.

You can provide this timestamp, but it isn't required upon the first observation. If you provide the field in this case, the timestamp should be the same as the `FirstObservedAt` timestamp. You should update this field to reflect the last or most recently observed timestamp each time a finding is observed.

Example

```
"LastObservedAt": "2017-03-23T13:22:13.933Z"
```

Malware (p. 189)

A list of malware related to a finding.

Example

```
"Malware": [  
  {  
    "Name": "Stringler",  
    "Type": "COIN_MINER",  
    "Path": "/usr/sbin/stringler",  
    "State": "OBSERVED"  
  }  
]
```

Network (p. 190) (Deprecated)

The details of network-related information about a finding.

This object is deprecated. To provide this data, either map the data to a resource in `Resources`, or use the `Action` object.

Example

```
"Network": {  
  "Direction": "IN",  
  "OpenPortRange": {  
    "Begin": 443,  
    "End": 443  
  },  
  "Protocol": "TCP",  
  "SourceIPv4": "1.2.3.4",  
  "SourceIPv6": "FE80:CD00:0000:0CDE:1257:0000:211E:729C",  
  "SourcePort": "42",  
  "SourceDomain": "example1.com",  
  "SourceMac": "00:0d:83:b1:c0:8e",  
  "DestinationIPv4": "2.3.4.5",  
  "DestinationIPv6": "FE80:CD00:0000:0CDE:1257:0000:211E:729C",  
  "DestinationPort": "80",  
}
```

```
"DestinationDomain": "example2.com"
}
```

NetworkPath (p. 190)

A network path that is related to the finding.

Each entry in `NetworkPath` represents a component of the path.

Note (p. 191)

A user-defined note that is added to a finding.

A finding provider can provide an initial note for a finding, but cannot add notes after that.

A note can only be updated using [BatchUpdateFindings](#).

Example

```
"Note": {
  "Text": "Don't forget to check under the mat.",
  "UpdatedBy": "jsmith",
  "UpdatedAt": "2018-08-31T00:15:09Z"
}
```

PatchSummary (p. 191)

Provides a summary of patch compliance.

Process (p. 192)

The details of process-related information about a finding.

Example:

```
"Process": {
  "Name": "syslogd",
  "Path": "/usr/sbin/syslogd",
  "Pid": 12345,
  "ParentPid": 56789,
  "LaunchedAt": "2018-09-27T22:37:31Z",
  "TerminatedAt": "2018-09-27T23:37:31Z"
}
```

ProductFields

A data type where security findings products can include additional solution-specific details that are not part of the defined AWS Security Finding Format.

For findings generated by Security Hub controls, `ProductFields` includes information about the control. See [the section called "Generating and updating control findings" \(p. 325\)](#).

This field should not contain redundant data and must not contain data that conflicts with AWS Security Finding Format fields.

The `aws/` prefix represents a reserved namespace for AWS products and services only and must not be submitted with findings from third-party integrations.

Although not required, products should format field names as `company-id/product-id/field-name`, where the `company-id` and `product-id` match those supplied in the `ProductArn` of the finding.

Field names can include the following characters: the following characters: A-Z, a-z, 0-9, blank spaces, and . : + \ = @ _ / - (hyphen).

Example

```
"ProductFields": {
  "generico/secure-pro/Count": "6",
  "generico/secure-pro/Action.Type", "AWS_API_CALL",
  "API", "DeleteTrail",
  "Service_Name": "cloudtrail.amazonaws.com",
  "aws/inspector/AssessmentTemplateName": "My daily CVE assessment",
  "aws/inspector/AssessmentTargetName": "My prod env",
  "aws/inspector/RulesPackageName": "Common Vulnerabilities and Exposures"
}
```

ProductName

The name of the product that generated the finding. For control-based findings, the product name is Security Hub.

Security Hub populates this attribute automatically for each finding. You cannot update it using [BatchImportFindings](#) or [BatchUpdateFindings](#). The exception to this is when you use a custom integration. See [the section called "Using custom product integrations" \(p. 321\)](#).

When you use the Security Hub console to filter findings by product name, you use this attribute.

When you use the Security Hub API to filter findings by product name, you use the `aws/securityhub/ProductName` attribute under `ProductFields`.

Security Hub does not synchronize those two attributes.

RecordState

The record state of a finding.

By default, when initially generated by a service, findings are considered `ACTIVE`.

The `ARCHIVED` state indicates that a finding should be hidden from view. Archived findings are not immediately deleted. You can search, review, and report on them.

`RecordState` is intended for finding providers, and can only be updated by [BatchImportFindings](#). It cannot be updated using [BatchUpdateFindings](#).

To track the status of your investigation into a finding, do not use `RecordState`. Instead, use [Workflow \(p. 274\)](#).

Finding providers update the record state. Security Hub also automatically archives control-based findings if the associated resource is deleted, the resource does not exist, or the control is disabled.

If the record state changes from `ARCHIVED` to `ACTIVE`, and the workflow status of the finding is either `NOTIFIED` or `RESOLVED`, then Security Hub automatically sets the workflow status to `NEW`.

Example

```
"RecordState": "ACTIVE"
```

Region

The AWS Region from which the finding was generated.

Security Hub populates this attribute automatically for each finding. You cannot update it using [BatchImportFindings](#) or [BatchUpdateFindings](#).

RelatedFindings (p. 192)

A list of related findings.

RelatedFindings should only be updated using [BatchUpdateFindings](#). It should not be updated using [BatchImportFindings](#).

To provide a list of related findings, finding providers should use the RelatedFindings object under [FindingProviderFields](#) (p. 189). See the section called "Using FindingProviderFields" (p. 120).

Example

```
"RelatedFindings": [  
  { "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",  
    "Id": "123e4567-e89b-12d3-a456-426655440000" },  
  { "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",  
    "Id": "AcmeNerfHerder-111111111111-x189dx7824" }  
]
```

Remediation (p. 192)

The remediation options for a finding.

Example

```
"Remediation": {  
  "Recommendation": {  
    "Text": "Run sudo yum update and cross your fingers and toes.",  
    "Url": "http://myfp.com/recommendations/  
dangerous_things_and_how_to_fix_them.html"  
  }  
}
```

Sample

Whether the finding is a sample finding.

SourceUrl

A URL that links to a page about the current finding in the finding product.

ThreatIntelIndicators (p. 273)

Threat intelligence details that are related to a finding.

Example

```
"ThreatIntelIndicators": [  
  {  
    "Type": "IPV4_ADDRESS",  
    "Value": "8.8.8.8",  
    "Category": "BACKDOOR",  
    "LastObservedAt": "2018-09-27T23:37:31Z",  
    "Source": "Threat Intel Weekly",  
    "SourceUrl": "http://threatintelweekly.org/backdoors/8888"  
  }  
]
```

Threats (p. 273)

Provides details about the threat detected by a finding.

Example

```
"Threats": [{
  "FilePaths": [{
    "FileName": "b.txt",
    "FilePath": "/tmp/b.txt",
    "Hash": "sha256",
    "ResourceId": "arn:aws:ec2:us-west-2:123456789012:volume/
vol-032f3bdd89aee112f",
  }],
  "ItemCount": 3,
  "Name": "Iot.linux.mirai.vwisi",
  "Severity": "HIGH"
}]
```

UserDefinedFields

A list of name-value string pairs that are associated with the finding. These are custom, user-defined fields that are added to a finding. These fields can be generated automatically through your specific configuration.

Findings products must not use this field for data that the product generates. Instead, findings products can use the `ProductFields` field for data that does not map to any standard AWS Security Finding Format field.

These fields can only be updated using [BatchUpdateFindings](#).

Example

```
"UserDefinedFields": {
  "reviewedByCio": "true",
  "comeBackToLater": "Check this again on Monday"
}
```

VerificationState

The veracity of a finding. Findings products can provide the value of `UNKNOWN` for this field. A findings product should provide this value if there is a meaningful analog in the findings product's system. This field is typically populated by a user determination or action after they investigate a finding.

A finding provider can provide an initial value for this attribute, but cannot update it after that. This attribute can only be updated using [BatchUpdateFindings](#).

[Vulnerabilities \(p. 273\)](#)

A list of vulnerabilities that apply to the finding.

[Workflow \(p. 274\)](#)

Provides information about the status of the investigation into a finding.

The workflow status is not intended for finding providers. The workflow status can only be updated using [BatchUpdateFindings](#). Customers can also update it from the console. See [the section called "Setting the workflow status for findings" \(p. 130\)](#).

Example

```
Workflow: {
  "Status": "NEW"
}
```

WorkflowState (Deprecated)

This field has been replaced by the `Status` field of the `Workflow` object.

The workflow state of a finding. Findings products can provide the value of `NEW` for this field. A findings product can provide a value for this field if there is a meaningful analog in the findings product's system.

Example

```
"WorkflowState": "NEW"
```

Action

The `Action` object provides details about an action that affects or that was taken on a resource. The action can be one of the following:

- A remote IP address issued an AWS API call.
- A DNS request was received.
- A remote IP address attempted to connect to an EC2 instance.
- A remote IP address attempted a port probe on an EC2 instance.

To view descriptions of `Action` attributes, see [Action](#) in the *AWS Security Hub API Reference*.

Example

```
"Action": {
  "ActionType": "PORT_PROBE",
  "PortProbeAction": {
    "PortProbeDetails": [
      {
        "LocalPortDetails": {
          "Port": 80,
          "PortName": "HTTP"
        },
        "LocalIpDetails": {
          "IpAddressV4": "192.0.2.0"
        },
        "RemoteIpDetails": {
          "Country": {
            "CountryName": "Example Country"
          },
          "City": {
            "CityName": "Example City"
          },
          "GeoLocation": {
            "Lon": 0,
            "Lat": 0
          },
          "Organization": {
            "AsnOrg": "ExampleASO",
            "Org": "ExampleOrg",
            "Isp": "ExampleISP",
            "Asn": 64496
          }
        }
      }
    ]
  },
  "Blocked": false
}
```

```
}
}
```

Compliance

Contains finding details related to a control. Only returned for findings that are generated as the result of a check that is run on a control.

To view descriptions of Compliance attributes, see [Compliance](#) in the *AWS Security Hub API Reference*.

Example

```
"Compliance": {
  "RelatedRequirements": ["Req1", "Req2"],
  "Status": "FAILED",
  "StatusReasons": [
    {
      "ReasonCode": "CLOUDWATCH_ALARMS_NOT_PRESENT",
      "Description": "CloudWatch alarms do not exist in the account"
    }
  ]
}
```

FindingProviderFields

In a [BatchImportFindings](#) request, finding providers use `FindingProviderFields` to provide values for attributes that should only be updated by [BatchUpdateFindings](#).

For details on how Security Hub handles updates from [BatchImportFindings](#) to `FindingProviderFields` and to the corresponding top-level attributes, see [the section called "Using FindingProviderFields" \(p. 120\)](#).

To view descriptions of `FindingProviderFields` attributes, see [FindingProviderFields](#) in the *AWS Security Hub API Reference*.

Example

```
"FindingProviderFields": {
  "Confidence": 42,
  "Criticality": 99,
  "RelatedFindings": [
    {
      "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
      "Id": "123e4567-e89b-12d3-a456-426655440000"
    }
  ],
  "Severity": {
    "Label": "MEDIUM",
    "Original": "MEDIUM"
  },
  "Types": [ "Software and Configuration Checks/Vulnerabilities/CVE" ]
}
```

Malware

The `Malware` object provides a list of malware that is related to a finding. It is an array that can contain up to five malware objects.

To view descriptions of Malware attributes, see [Malware](#) in the *AWS Security Hub API Reference*.

Example

```
"Malware": [
  {
    "Name": "Stringler",
    "Type": "COIN_MINER",
    "Path": "/usr/sbin/stringler",
    "State": "OBSERVED"
  }
]
```

Network (Deprecated)

An example of network-related information about a finding.

This object is deprecated. To provide this data, you can either map it to a resource in `Resources`, or use the `Action` object.

To view descriptions of `Network` attributes, see [Network](#) in the *AWS Security Hub API Reference*.

Example

```
"Network": {
  "Direction": "IN",
  "OpenPortRange": {
    "Begin": 443,
    "End": 443
  },
  "Protocol": "TCP",
  "SourceIPv4": "1.2.3.4",
  "SourceIPv6": "FE80:CD00:0000:0CDE:1257:0000:211E:729C",
  "SourcePort": "42",
  "SourceDomain": "example1.com",
  "SourceMac": "00:0d:83:b1:c0:8e",
  "DestinationIPv4": "2.3.4.5",
  "DestinationIPv6": "FE80:CD00:0000:0CDE:1257:0000:211E:729C",
  "DestinationPort": "80",
  "DestinationDomain": "example2.com"
}
```

NetworkPath

The `NetworkPath` object provides information about a network path that is relevant to a finding. Each entry under `NetworkPath` represents a component of that path.

To view descriptions of `NetworkPath` attributes, see [NetworkPathComponent](#) in the *AWS Security Hub API Reference*.

Example

```
"NetworkPath" : [
  {
    "ComponentId": "abc-01a234bc56d8901ee",
    "ComponentType": "AWS::EC2::InternetGateway",
    "Egress": {
      "Destination": {
        "Address": [ "192.0.2.0/24" ],
        "PortRanges": [
          {
            "Begin": 443,
```

```

        "End": 443
      }
    ]
  },
  "Protocol": "TCP",
  "Source": {
    "Address": ["203.0.113.0/24"]
  }
},
"Ingress": {
  "Destination": {
    "Address": [ "198.51.100.0/24" ],
    "PortRanges": [
      {
        "Begin": 443,
        "End": 443
      }
    ]
  },
  "Protocol": "TCP",
  "Source": {
    "Address": [ "203.0.113.0/24" ]
  }
}
}
]

```

Note

The Note object adds a user-defined note to the finding.

A finding provider can provide an initial note for a finding but cannot add notes after that. A note can only be updated using [BatchUpdateFindings](#).

To view descriptions of Note attributes, see [Note](#) in the *AWS Security Hub API Reference*.

Example

```

"Note": {
  "Text": "Don't forget to check under the mat.",
  "UpdatedBy": "jsmith",
  "UpdatedAt": "2018-08-31T00:15:09Z"
}

```

PatchSummary

The PatchSummary object provides an overview of the patch compliance status for an instance against a selected compliance standard.

To view descriptions of PatchSummary attributes, see [PatchSummary](#) in the *AWS Security Hub API Reference*.

Example

```

"PatchSummary" : {
  "Id" : "pb-123456789098"
  "InstalledCount" : "100",
  "MissingCount" : "100",
  "FailedCount" : "0",
  "InstalledOtherCount" : "1023",
  "InstalledRejectedCount" : "0",

```

```
"InstalledPendingReboot" : "0",
"OperationStartTime" : "2018-09-27T23:37:31Z",
"OperationEndTime" : "2018-09-27T23:39:31Z",
"RebootOption" : "RebootIfNeeded",
"Operation" : "Install"
}
```

Process

The `Process` object provides process-related details about the finding.

To view descriptions of `Process` attributes, see [ProcessDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"Process": {
  "Name": "syslogd",
  "Path": "/usr/sbin/syslogd",
  "Pid": 12345,
  "ParentPid": 56789,
  "LaunchedAt": "2018-09-27T22:37:31Z",
  "TerminatedAt": "2018-09-27T23:37:31Z"
}
```

RelatedFindings

The `RelatedFindings` object provides a list of findings that are related to the current finding.

For [BatchImportFindings](#) requests, finding providers should use the `RelatedFindings` object under [FindingProviderFields](#) (p. 189).

To view descriptions of `RelatedFindings` attributes, see [RelatedFinding](#) in the *AWS Security Hub API Reference*.

Example

```
"RelatedFindings": [
  { "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
    "Id": "123e4567-e89b-12d3-a456-426655440000" },
  { "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/guardduty",
    "Id": "AcmeNerfHerder-111111111111-x189dx7824" }
]
```

Remediation

The `Remediation` object provides information about recommended remediation steps to address the finding.

To view descriptions of `Remediation` attributes, see [Remediation](#) in the *AWS Security Hub API Reference*.

Example

```
"Remediation": {
  "Recommendation": {
    "Text": "Run sudo yum update and cross your fingers and toes.",
    "Url": "http://myfp.com/recommendations/dangerous_things_and_how_to_fix_them.html"
  }
}
```


Resources

The `Resources` object provides information about the resources involved in a finding.

It contains an array of up to 32 resource objects.

Example

```
"Resources": [
  {
    "Type": "AwsEc2Instance",
    "Id": "arn:aws:ec2:us-west-2:111122223333:instance/i-1234567890abcdef0",
    "Partition": "aws",
    "Region": "us-west-2",
    "ResourceRole": "Target",
    "Tags": {
      "billingCode": "Lotus-1-2-3",
      "needsPatching": "true"
    },
    "Details": {
      "AwsEc2Instance": {
        "Type": "i3.xlarge",
        "ImageId": "ami-abcd1234",
        "IPv4Addresses": [ "54.194.252.215", "192.168.1.88" ],
        "IPv6Addresses": [ "2001:db8:1234:1a2b::123" ],
        "KeyName": "my_keypair",
        "IamInstanceProfileArn": "arn:aws:iam::111111111111:instance-profile/AdminRole",
        "VpcId": "vpc-11112222",
        "SubnetId": "subnet-56f5f633",
        "LaunchedAt": "2018-05-08T16:46:19.000Z"
      }
    }
  }
]
```

Topics

- [Resource attributes \(p. 195\)](#)
- [DataClassification \(p. 199\)](#)
- [AwsApiGatewayRestApi \(p. 208\)](#)
- [AwsApiGatewayStage \(p. 208\)](#)
- [AwsApiGatewayV2Api \(p. 209\)](#)
- [AwsApiGatewayV2Stage \(p. 210\)](#)
- [AwsAutoScalingAutoScalingGroup \(p. 211\)](#)
- [AwsAutoScalingLaunchConfiguration \(p. 212\)](#)
- [AwsCertificateManagerCertificate \(p. 213\)](#)
- [AwsCloudFormationStack \(p. 215\)](#)
- [AwsCloudFrontDistribution \(p. 215\)](#)
- [AwsCloudTrailTrail \(p. 217\)](#)
- [AwsCloudWatchAlarm \(p. 217\)](#)
- [AwsCodeBuildProject \(p. 218\)](#)
- [AwsDynamoDbTable \(p. 219\)](#)
- [AwsEc2Eip \(p. 222\)](#)
- [AwsEc2Instance \(p. 222\)](#)
- [AwsEc2NetworkAcl \(p. 223\)](#)
- [AwsEc2NetworkInterface \(p. 223\)](#)

- [AwsEc2SecurityGroup](#) (p. 224)
- [AwsEc2Subnet](#) (p. 224)
- [AwsEc2TransitGateway](#) (p. 225)
- [AwsEc2Volume](#) (p. 225)
- [AwsEc2Vpc](#) (p. 226)
- [AwsEc2VpcEndpointService](#) (p. 227)
- [AwsEc2VpcPeeringConnection](#) (p. 227)
- [AwsEc2VpnConnection](#) (p. 228)
- [AwsEcrContainerImage](#) (p. 229)
- [AwsEcrRepository](#) (p. 230)
- [AwsEcsCluster](#) (p. 230)
- [AwsEcsContainer](#) (p. 231)
- [AwsEcsService](#) (p. 232)
- [AwsEcsTask](#) (p. 233)
- [AwsEcsTaskDefinition](#) (p. 234)
- [AwsEfsAccessPoint](#) (p. 236)
- [AwsEksCluster](#) (p. 237)
- [AwsElasticBeanstalkEnvironment](#) (p. 238)
- [AwsElasticSearchDomain](#) (p. 239)
- [AwsElbLoadBalancer](#) (p. 240)
- [AwsElbv2LoadBalancer](#) (p. 242)
- [AwsIamAccessKey](#) (p. 243)
- [AwsIamGroup](#) (p. 243)
- [AwsIamPolicy](#) (p. 244)
- [AwsIamRole](#) (p. 244)
- [AwsIamUser](#) (p. 245)
- [AwsKinesisStream](#) (p. 246)
- [AwsKmsKey](#) (p. 246)
- [AwsLambdaFunction](#) (p. 246)
- [AwsLambdaLayerVersion](#) (p. 247)
- [AwsNetworkFirewallFirewall](#) (p. 248)
- [AwsNetworkFirewallFirewallPolicy](#) (p. 248)
- [AwsNetworkFirewallRuleGroup](#) (p. 249)
- [AwsOpenSearchServiceDomain](#) (p. 251)
- [AwsRdsDbCluster](#) (p. 252)
- [AwsRdsDbClusterSnapshot](#) (p. 254)
- [AwsRdsDbInstance](#) (p. 254)
- [AwsRdsDbSecurityGroup](#) (p. 256)
- [AwsRdsDbSnapshot](#) (p. 257)
- [AwsRdsEventSubscription](#) (p. 258)
- [AwsRedshiftCluster](#) (p. 258)
- [AwsS3AccountPublicAccessBlock](#) (p. 261)
- [AwsS3Bucket](#) (p. 262)
- [AwsS3Object](#) (p. 263)
- [AwsSecretsManagerSecret](#) (p. 264)

- [AwsSnsTopic](#) (p. 264)
- [AwsSqsQueue](#) (p. 265)
- [AwsSsmPatchCompliance](#) (p. 265)
- [AwsWafRateBasedRule](#) (p. 266)
- [AwsWafRegionalRateBasedRule](#) (p. 266)
- [AwsWafRegionalRule](#) (p. 267)
- [AwsWafRegionalRuleGroup](#) (p. 267)
- [AwsWafRegionalWebAcl](#) (p. 268)
- [AwsWafRule](#) (p. 268)
- [AwsWafRuleGroup](#) (p. 269)
- [AwsWafWebAcl](#) (p. 269)
- [AwsXrayEncryptionConfig](#) (p. 270)
- [Container](#) (p. 270)
- [Other](#) (p. 270)

Resource attributes

Here are details and examples for the `Resources` object in the AWS Security Finding Format (ASFF). To view attributes for the `Resources` object, see [Resource](#) in the *AWS Security Hub API Reference*.

DataClassification (p. 199)

Used to provide details about sensitive data that was detected on a resource.

Details

This field provides additional details about a single resource using the appropriate objects.

Note that if the finding size exceeds the maximum of 240 KB, then the `Details` object is removed from the finding. For findings for controls that use AWS Config rules, you can view the resource details on the AWS Config console.

Each resource must be provided in a separate resource object in the `Resources` object.

Security Hub provides a set of available resource details objects for its supported resource types. These objects correspond to values of the resource `Type`. Use the provided types and objects whenever possible.

For example, if the resource is an S3 bucket, then set the resource `Type` to `AwsS3Bucket` and provide the resource details in the [AwsS3Bucket](#) (p. 262) object.

The [Other](#) (p. 270) object allows you to provide custom fields and values. You use the [Other](#) (p. 270) object in the following cases.

- The resource type (the value of the resource `Type`) does not have a corresponding details object. To provide details for the resource, you use the [Other](#) (p. 270) details object.
- The object for the resource type does not include all of the fields that you want to populate. In this case, use the details object for the resource type to populate the available fields. Use the `Other` object to populate the fields that are not in the type-specific object.
- The resource type is not one of the provided types. In this case, set the resource `Type` to `Other`, and use the [Other](#) (p. 270) details object to populate the details.

Example

```
"Details": {
```

```
"AwsEc2Instance": {
  "IamInstanceProfileArn": "arn:aws:iam::123456789012:role/IamInstanceProfileArn",
  "ImageId": "ami-79fd7eee",
  "IPv4Addresses": ["1.1.1.1"],
  "IPv6Addresses": ["2001:db8:1234:1a2b::123"],
  "KeyName": "testkey",
  "LaunchedAt": "2018-09-29T01:25:54Z",
  "MetadataOptions": {
    "HttpEndpoint": "enabled",
    "HttpProtocolIpv6": "enabled",
    "HttpPutResponseHopLimit": 1,
    "HttpTokens": "optional",
    "InstanceMetadataTags": "disabled"
  },
  "NetworkInterfaces": [
    {
      "NetworkInterfaceId": "eni-e5aa89a3"
    }
  ],
  "SubnetId": "PublicSubnet",
  "Type": "i3.xlarge",
  "VirtualizationType": "hvm",
  "VpcId": "TestVPCIPv6"
},
"AwsS3Bucket": {
  "OwnerId": "da4d66eac431652a4d44d490a00500bde52c97d235b7b4752f9f688566fe6de",
  "OwnerName": "acmes3bucketowner"
},
"Other": { "LightPen": "blinky", "SerialNo": "1234abcd" }
}
```

Id

The canonical identifier for the given resource type.

For AWS resources that are identified by ARNs, this must be the ARN.

For all other AWS resource types that lack ARNs, this must be the identifier as defined by the AWS service that created the resource.

For non AWS resources, this should be a unique identifier that is associated with the resource.

Example

```
"Id": "arn:aws:s3:::example-bucket"
```

Partition

The canonical AWS partition name that the Region is assigned to.

Example

```
"Partition": "aws"
```

Region

The canonical AWS external Region name where this resource is located.

Example

```
"Region": "us-west-2"
```

ResourceRole

Identifies the role of the resource in the finding. A resource is either the target of the finding activity or the actor that performed the activity.

Tags

A list of AWS tags that are associated with a resource at the time the finding was processed. Include the `Tags` attribute only for resources that have an associated tag. If a resource has no associated tag, don't include a `Tags` attribute in the finding.

The following basic restrictions apply to tags:

- You can provide only tags that actually exist on an AWS resource in this field. To provide data for a resource type that isn't defined in the AWS Security Finding Format, use the `Other` details subfield.
- Values are limited to the following characters: A-Z, a-z, 0-9, blank spaces, and `. : + = @ _ / -` (hyphen).
- Values are limited to the AWS tag value length of 256 characters max.

Example

```
"Tags": {  
  "billingCode": "Lotus-1-2-3",  
  "needsPatching": "true"  
}
```

Type

The type of the resource that you are providing details for.

Whenever possible, use one of the provided resource types, such as `AwsEc2Instance` or `AwsS3Bucket`.

If the resource type does not match any of the provided resource types, then set the resource `Type` to `Other`, and use the `Other` details subfield to populate the details.

Supported values are as follows. If a type has a corresponding `Details` object, then to view the details for the `Details` object, choose the type name.

- `AwsAccount`
- `AwsApiGatewayMethod`
- [AwsApiGatewayRestApi \(p. 208\)](#)
- [AwsApiGatewayStage \(p. 208\)](#)
- [AwsApiGatewayV2Api \(p. 209\)](#)
- [AwsApiGatewayV2Stage \(p. 210\)](#)
- `AwsAppStreamFleet`
- [AwsAutoscalingAutoscalingGroup \(p. 211\)](#)
- [AwsAutoscalingLaunchConfiguration \(p. 212\)](#)
- [AwsCertificateManagerCertificate \(p. 213\)](#)
- `AwsCloudFormationStack`
- [AwsCloudFrontDistribution \(p. 215\)](#)
- [AwsCloudTrailTrail \(p. 217\)](#)
- `AwsCloudWatchAlarm`
- [AwsCodeBuildProject \(p. 218\)](#)
- `AwsCodeCommitRepository`

- [AwsCodeDeployApplication](#)
- [AwsCodeDeployDeploymentGroup](#)
- [AwsCodePipelinePipeline](#)
- [AwsCognitoIdentityPool](#)
- [AwsCognitoUserPool](#)
- [AwsDmsReplicationInstance](#)
- [AwsDynamoDbTable](#) (p. 219)
- [AwsEc2Eip](#) (p. 222)
- [AwsEc2Instance](#) (p. 222)
- [AwsEc2NetworkAcl](#) (p. 223)
- [AwsEc2NetworkInterface](#) (p. 223)
- [AwsEc2SecurityGroup](#) (p. 224)
- [AwsEc2Snapshot](#)
- [AwsEc2Subnet](#) (p. 224)
- [AwsEc2Volume](#) (p. 225)
- [AwsEc2Vpc](#) (p. 226)
- [AwsEc2VpnConnection](#) (p. 228)
- [AwsEc2VpcEndPointService](#) (p. 227)
- [AwsEcrContainerImage](#) (p. 229)
- [AwsEcrRepository](#) (p. 230)
- [AwsEcsCluster](#) (p. 230)
- [AwsEcsService](#) (p. 232)
- [AwsEcsTaskDefinition](#) (p. 234)
- [AwsEfsFileSystem](#)
- [AwsEksCluster](#) (p. 237)
- [AwsElasticCacheCacheCluster](#)
- [AwsElasticBeanstalkEnvironment](#) (p. 238)
- [AwsElasticsearchDomain](#) (p. 239)
- [AwsElbLoadBalancer](#) (p. 240)
- [AwsElbv2LoadBalancer](#) (p. 242)
- [AwsEmrCluster](#)
- [AwsIamAccessKey](#) (p. 243)
- [AwsIamGroup](#) (p. 243)
- [AwsIamPolicy](#) (p. 244)
- [AwsIamRole](#) (p. 244)
- [AwsIamUser](#) (p. 245)
- [AwsKinesisStream](#)
- [AwsKmsKey](#) (p. 246)
- [AwsLambdaFunction](#) (p. 246)
- [AwsLambdaLayerVersion](#) (p. 247)
- [AwsLogsLogGroup](#)
- [AwsNetworkFirewallFirewall](#) (p. 248)
- [AwsNetworkFirewallFirewallPolicy](#) (p. 248)
- [AwsNetworkFirewallRuleGroup](#) (p. 249)
- [AwsOpenSearchServiceDomain](#) (p. 251)

- [AwsRdsDbCluster](#) (p. 252)
- [AwsRdsDbClusterSnapshot](#) (p. 254)
- [AwsRdsDbInstance](#) (p. 254)
- [AwsRdsDbSnapshot](#) (p. 257)
- [AwsRdsEventSubscription](#) (p. 258)
- [AwsRedshiftCluster](#) (p. 258)
- [AwsS3AccountPublicAccessBlock](#) (p. 261)
- [AwsS3Bucket](#) (p. 262)
- [AwsS3Object](#) (p. 263)
- [AwsSageMakerNotebookInstance](#)
- [AwsSecretsManagerSecret](#) (p. 264)
- [AwsSnsTopic](#) (p. 264)
- [AwsSqsQueue](#) (p. 265)
- [AwsSsmAssociationCompliance](#)
- [AwsSsmPatchCompliance](#) (p. 265)
- [AwsWafRateBasedRule](#) (p. 266)
- [AwsWafRegionalRateBasedRule](#) (p. 266)
- [AwsWafWebAcl](#) (p. 269)
- [AwsXrayEncryptionConfig](#) (p. 270)
- [Container](#) (p. 270)
- [Other](#) (p. 270)

Example

```
"Type": "AwsS3Bucket"
```

DataClassification

The `DataClassification` object contains information about sensitive data that was detected on the resource.

Example

```
"DataClassification": {
  "DetailedResultsLocation": "Path_to_Folder_Or_File",
  "Result": {
    "MimeType": "text/plain",
    "SizeClassified": 2966026,
    "AdditionalOccurrences": false,
    "Status": {
      "Code": "COMPLETE",
      "Reason": "Unsupportedfield"
    },
  },
  "SensitiveData": [
    {
      "Category": "PERSONAL_INFORMATION",
      "Detections": [
        {
          "Count": 34,
          "Type": "GE_PERSONAL_ID",
          "Occurrences": {
            "LineRanges": [
              {
```

```

        "Start": 1,
        "End": 10,
        "StartColumn": 20
    }
    ],
    "Pages": [],
    "Records": [],
    "Cells": []
}
},
{
    "Count": 59,
    "Type": "EMAIL_ADDRESS",
    "Occurrences": {
        "Pages": [
            {
                "PageNumber": 1,
                "OffsetRange": {
                    "Start": 1,
                    "End": 100,
                    "StartColumn": 10
                },
                "LineRange": {
                    "Start": 1,
                    "End": 100,
                    "StartColumn": 10
                }
            }
        ]
    }
},
{
    "Count": 2229,
    "Type": "URL",
    "Occurrences": {
        "LineRanges": [
            {
                "Start": 1,
                "End": 13
            }
        ]
    }
},
{
    "Count": 13826,
    "Type": "NameDetection",
    "Occurrences": {
        "Records": [
            {
                "RecordIndex": 1,
                "JsonPath": "$.ssn.value"
            }
        ]
    }
},
{
    "Count": 32,
    "Type": "AddressDetection"
}
],
"TotalCount": 32
}
],
"CustomDataIdentifiers": {
    "Detections": [
        {

```



```
        "Arn": "1712be25e7c7f53c731fe464f1c869b8",  
        "Name": "1712be25e7c7f53c731fe464f1c869b8",  
        "Count": 2,  
      }  
    ],  
    "TotalCount": 2  
  }  
}
```

DataClassification attributes

DataClassification can contain the following attributes.

DetailedResultsLocation

Optional

The path to the folder or file that contains the sensitive data.

Type: String

[Result \(p. 201\)](#)

Optional

The details about the sensitive data that was detected on the resource.

Type: Object

Result

The Result object contains the details about the sensitive data that was detected on the resource.

Result can have the following attributes.

AdditionalOccurrences

Optional

Indicates whether there are additional occurrences of sensitive data that are not included in the finding. This occurs when the number of occurrences exceeds the maximum that can be included.

Type: Boolean

[CustomDataIdentifiers \(p. 202\)](#)

Optional

Provides details about sensitive data that was identified based on customer-defined configuration.

Type: Object

MimeType

Optional

The type of content that the finding applies to.

Type: String

Required format: Must be a MIME type.

Examples

- `application/gzip`, for a GNU Gzip compressed archive file
- `application/pdf`, for an Adobe Portable Document Format (.pdf) file

[SensitiveData \(p. 203\)](#)

Optional

Provides details about sensitive data that was identified based on built-in configuration.

Type: Object

`SizeClassified`

Optional

The total size in bytes of the affected data.

Type: Long

`Status`

Optional

Provides details about the current status of the detection.

Type: Object

`Status.Code`

Optional

The code that represents the status of the sensitive data detection.

Type: String

`Status.Reason`

Optional

A longer description of the current status of the sensitive data detection.

Type: String

[CustomDataIdentifiers](#)

`CustomDataIdentifiers` contains instances of sensitive data that were detected by user-defined identifiers.

`CustomDataIdentifiers` can have the following attributes.

[Detections \(p. 203\)](#)

Optional

The list of detected instances of sensitive data.

Type: Array of objects

`TotalCount`

Optional

The total number of occurrences of sensitive data.

Type: Integer

SensitiveData

`SensitiveData` contains detected instances of sensitive data that are based on built-in identifiers.

Category

Optional

The category of sensitive data that was detected. For example, the category might indicate that the sensitive data involved credentials, financial information, or personal information.

Type: String

[Detections \(p. 203\)](#)

Optional

The list of detected instances of sensitive data.

Type: Array of objects

TotalCount

Optional

The total number of occurrences of sensitive data.

Type: Integer

Detections

`Detections` contains the details of the sensitive data that was detected.

Each instance of detected sensitive data can have the following attributes.

Arn

Optional

The ARN of the custom identifier that was used to detect the sensitive data.

`Arn` is only provided in `Detections` instances under `CustomDataIdentifiers`. `Detections` instances under `SensitiveData` do not have the `Arn` attribute.

Type: String

Count

Optional

The total number of occurrences of sensitive data that were detected.

Type: Integer

Name

Optional

The name of the custom identifier that detected the sensitive data.

`Name` is only provided in `Detections` instances under `CustomDataIdentifiers`. `Detections` instances under `SensitiveData` do not have the `Name` attribute.

Type: String

Occurrences

Optional

Details about the sensitive data that was detected. `Occurrences` can contain the following objects. Each of these objects contains an array of objects.

- [Cells \(p. 204\)](#) – Contains occurrences of sensitive data detected in Microsoft Excel files, comma-separated value (.csv) files, or tab-separated value (.tsv) files
- [LineRanges \(p. 205\)](#) – Contains occurrences of sensitive data detected in nonbinary text files or Microsoft Word files
- [OffsetRanges \(p. 205\)](#) – Contains occurrences of sensitive data detected in binary text files
- [Pages \(p. 206\)](#) – Contains occurrences of sensitive data detected in Adobe Portable Document Format (.pdf) files
- [Records \(p. 207\)](#) – Contains occurrences of sensitive data detected in an Apache Avro object container or an Apache Parquet file

Type: Object

Type

Optional

The type of sensitive data that was detected. For example, the type might indicate that the data is an email address.

Type is only provided in `Detections` instances under `SensitiveData.Detections` instances under `CustomDataIdentifiers` do not have the `Type` attribute.

Type: String

Cells

`Cells` contains occurrences of sensitive data detected in Microsoft Excel workbooks, comma-separated value (.csv) files, or tab-separated value (.tsv) files.

For each occurrence, `Cells` identifies the cell that contains the sensitive data.

Each occurrence can contain the following attributes.

`CellReference`

Optional

For a Microsoft Excel workbook, provides the location of the cell, as an absolute cell reference, that contains the data. For example, `Sheet2!C5` for cell C5 on Sheet2.

This attribute is null for .csv and .tsv files.

Type: String

`Column`

Optional

The column number of the column that contains the data. For a Microsoft Excel workbook, the column number corresponds to the alphabetical column identifiers. For example, a value of 1 for `Column` corresponds to the A column in the workbook.

Type: Integer

ColumnName

Optional

The name of the column that contains the data.

Type: String

Row

Optional

The row number of the row that contains the data.

Type: Integer

LineRanges

`LineRanges` contains occurrences of sensitive data detected in a nonbinary text file or a Microsoft Word file. Nonbinary text files include files such as `.html`, `.xml`, `.json`, and `.txt` files.

For each occurrence, `LineRanges` provides the location of the sensitive data. The location includes the line where the data is located and the position of the data on that line.

Each occurrence can have the following attributes.

End

Optional

The number of lines from the beginning of the file to the end of the sensitive data.

Type: Integer

Start

Optional

The number of lines from the beginning of the file to the end of the sensitive data.

Type: Integer

StartColumn

Optional

In the line where the sensitive data begins, the column within the line where the sensitive data starts.

Type: Integer

OffsetRanges

`OffsetRanges` contains occurrences of sensitive data detected in a binary text file.

For each occurrence, `OffsetRanges` provides the location of the sensitive data. The location is the number of characters relative to the beginning of the file.

Each occurrence can have the following attributes.

End

Optional

The number of characters from the beginning of the file to the end of the sensitive data.

Type: Integer

Start

Optional

The number of characters from the beginning of the file to the beginning of the sensitive data.

Type: Integer

StartColumn

Optional

The column where the sensitive data begins.

Type: Integer

Pages

Pages provides occurrences of sensitive data in an Adobe Portable Document Format (.pdf) file.

For each occurrence, Pages identifies the page that contains the data, and provides the position of the data on the page. The position can be identified using either line numbers or numbers of characters.

Each occurrence can have the following attributes.

LineRange

Optional

Uses line numbers to identify the location of the sensitive data.

Type: Object

LineRange.End

Optional

The number of lines from the beginning of the file to the end of the sensitive data.

Type: Integer

LineRange.Start

Optional

The number of lines from the beginning of the file to the beginning of the sensitive data.

Type: Integer

LineRange.StartColumn

Optional

On the line where the sensitive data begins, the column number where the sensitive data begins.

Type: Integer

`OffsetRange`

Optional

Identifies the location of the sensitive data based on the number of characters.

Type: Object

`OffsetRange.End`

Optional

The number of characters from the beginning of the file to the end of the sensitive data.

Type: Integer

`OffsetRange.Start`

Optional

The number of characters from the beginning of the file to the beginning of the sensitive data.

Type: Integer

`OffsetRange.StartColumn`

Optional

On the line where the sensitive data starts, the column number where it starts.

Type: Integer

`PageNumber`

Optional

The page number of the page that contains the sensitive data.

Type: Integer

Records

`Records` identifies occurrences of sensitive data in an Apache Avro object container or an Apache Parquet file.

For each occurrence, `Records` specifies the record index and the path to the field that contains the sensitive data.

Each occurrence can have the following attributes.

`JsonPath`

Optional

The path, as a JSONPath expression, to the field in the record that contains the data. If the field name is longer than 20 characters, it is truncated. If the path is longer than 250 characters, it is truncated.

Type: String

`RecordIndex`

Optional

The record index, starting from 0, for the record that contains the data.

Type: Integer

AwsApiGatewayRestApi

The `AwsApiGatewayRestApi` object contains information about a REST API in version 1 of Amazon API Gateway.

The following is an example `AwsApiGatewayRestApi` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayRestApi` attributes, see [AwsApiGatewayRestApiDetails](#) in the *AWS Security Hub API Reference*.

Example

```
AwsApiGatewayRestApi: {
  "Id": "exampleapi",
  "Name": "Security Hub",
  "Description": "AWS Security Hub",
  "CreateDate": "2018-11-18T10:20:05-08:00",
  "Version": "2018-10-26",
  "BinaryMediaTypes" : ["-*~1*"],
  "MinimumCompressionSize": 1024,
  "ApiKeySource": "AWS_ACCOUNT_ID",
  "EndpointConfiguration": {
    "Types": [
      "REGIONAL"
    ]
  }
}
```

AwsApiGatewayStage

The `AwsApiGatewayStage` object provides information about a version 1 Amazon API Gateway stage.

The following is an example `AwsApiGatewayStage` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayStage` attributes, see [AwsApiGatewayStageDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsApiGatewayStage": {
  "DeploymentId": "n7hlmf",
  "ClientCertificateId": "a1b2c3",
  "StageName": "Prod",
  "Description" : "Stage Description",
  "CacheClusterEnabled": false,
  "CacheClusterSize" : "1.6",
  "CacheClusterStatus": "NOT_AVAILABLE",
  "MethodSettings": [
    {
      "MetricsEnabled": true,
      "LoggingLevel": "INFO",
      "DataTraceEnabled": false,
      "ThrottlingBurstLimit": 100,
      "ThrottlingRateLimit": 5.0,
      "CachingEnabled": false,
      "CacheTtlInSeconds": 300,
      "CacheDataEncrypted": false,
      "RequireAuthorizationForCacheControl": true,
      "UnauthorizedCacheControlHeaderStrategy": "SUCCEED_WITH_RESPONSE_HEADER",
      "HttpMethod": "POST",
      "ResourcePath": "/echo"
    }
  ]
}
```



```

    ],
    "Variables": {"test": "value"},
    "DocumentationVersion": "2.0",
    "AccessLogSettings": {
      "Format": "{\requestId\": \"\$context.requestId\", \"extendedRequestId
\": \"\$context.extendedRequestId\", \"ownerAccountId\": \"\$context.accountId\",
\": \"\$context.identity.accountId\", \"callerPrincipal\":
\": \"\$context.identity.caller\", \"httpMethod\": \"\$context.httpMethod\", \"resourcePath
\": \"\$context.resourcePath\", \"status\": \"\$context.status\", \"requestTime
\": \"\$context.requestTime\", \"responseLatencyMs\": \"\$context.responseLatency
\", \"errorMessage\": \"\$context.error.message\", \"errorResponseType\":
\": \"\$context.error.responseType\", \"apiId\": \"\$context.apiId\", \"awsEndpointRequestId
\": \"\$context.awsEndpointRequestId\", \"domainName\": \"\$context.domainName\", \"stage
\": \"\$context.stage\", \"xrayTraceId\": \"\$context.xrayTraceId\", \"sourceIp\":
\": \"\$context.identity.sourceIp\", \"user\": \"\$context.identity.user\", \"userAgent
\": \"\$context.identity.userAgent\", \"userArn\": \"\$context.identity.userArn\",
\": \"\$context.integrationLatency\", \"integrationLatency\": \"\$context.integrationLatency\", \"integrationStatus
\": \"\$context.integrationStatus\", \"authorizerIntegrationLatency\":
\": \"\$context.authorizer.integrationLatency\" }",
      "DestinationArn": "arn:aws:logs:us-west-2:111122223333:log-
group:SecurityHubAPIAccessLog/Prod"
    },
    "CanarySettings": {
      "PercentTraffic": 0.0,
      "DeploymentId": "ul73s8",
      "StageVariableOverrides": [
        "String": "String"
      ],
    },
    "UseStageCache": false
  },
  "TracingEnabled": false,
  "CreatedDate": "2018-07-11T10:55:18-07:00",
  "LastUpdatedDate": "2020-08-26T11:51:04-07:00",
  "WebAclArn": "arn:aws:waf-regional:us-west-2:111122223333:webacl/
cb606bd8-5b0b-4f0b-830a-dd304e48a822"
}

```

AwsApiGatewayV2Api

The `AwsApiGatewayV2Api` object contains information about a version 2 API in Amazon API Gateway.

The following is an example `AwsApiGatewayV2Api` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Api` attributes, see [AwsApiGatewayV2ApiDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsApiGatewayV2Api": {
  "ApiEndpoint": "https://example.us-west-2.amazonaws.com",
  "ApiId": "alb2c3d4",
  "ApiKeySelectionExpression": "$request.header.x-api-key",
  "CreatedDate": "2020-03-28T00:32:37Z",
  "Description": "ApiGatewayV2 Api",
  "Version": "string",
  "Name": "my-api",
  "ProtocolType": "HTTP",
  "RouteSelectionExpression": "$request.method $request.path",
  "CorsConfiguration": {
    "AllowOrigins": [ "*" ],
    "AllowCredentials": true,
    "ExposeHeaders": [ "string" ],
    "MaxAge": 3000,
    "AllowMethods": [
      "GET",

```

```

        "PUT",
        "POST",
        "DELETE",
        "HEAD"
    ],
    "AllowHeaders": [ "*" ]
}
}

```

AwsApiGatewayV2Stage

AwsApiGatewayV2Stage contains information about a version 2 stage for Amazon API Gateway.

The following is an example AwsApiGatewayV2Stage finding in the AWS Security Finding Format (ASFF). To view descriptions of AwsApiGatewayV2Stage attributes, see [AwsApiGatewayV2StageDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsApiGatewayV2Stage": {
  "CreateDate": "2020-04-08T00:36:05Z",
  "Description": "ApiGatewayV2",
  "DefaultRouteSettings": {
    "DetailedMetricsEnabled": false,
    "LoggingLevel": "INFO",
    "DataTraceEnabled": true,
    "ThrottlingBurstLimit": 100,
    "ThrottlingRateLimit": 50
  },
  "DeploymentId": "x1zwyv",
  "LastUpdatedDate": "2020-04-08T00:36:13Z",
  "RouteSettings": {
    "DetailedMetricsEnabled": false,
    "LoggingLevel": "INFO",
    "DataTraceEnabled": true,
    "ThrottlingBurstLimit": 100,
    "ThrottlingRateLimit": 50
  },
  "StageName": "prod",
  "StageVariables": [
    "function": "my-prod-function"
  ],
  "AccessLogSettings": {
    "Format": "{\requestId\": \"\$context.requestId\", \"extendedRequestId
\": \"\$context.extendedRequestId\", \"ownerAccountId\": \"\$context.accountId\",
\": \"\$context.identity.caller\", \"httpMethod\": \"\$context.httpMethod\", \"resourcePath
\": \"\$context.resourcePath\", \"status\": \"\$context.status\", \"requestTime
\": \"\$context.requestTime\", \"responseLatencyMs\": \"\$context.responseLatency
\", \"errorMessage\": \"\$context.error.message\", \"errorResponseType\":
\": \"\$context.error.responseType\", \"apiId\": \"\$context.apiId\", \"awsEndpointRequestId
\": \"\$context.awsEndpointRequestId\", \"domainName\": \"\$context.domainName\", \"stage
\": \"\$context.stage\", \"xrayTraceId\": \"\$context.xrayTraceId\", \"sourceIp\":
\": \"\$context.identity.sourceIp\", \"user\": \"\$context.identity.user\", \"userAgent
\": \"\$context.identity.userAgent\", \"userArn\": \"\$context.identity.userArn\",
\": \"\$context.integrationLatency\", \"integrationStatus
\": \"\$context.integrationStatus\", \"authorizerIntegrationLatency\":
\": \"\$context.authorizer.integrationLatency\" }",
    "DestinationArn": "arn:aws:logs:us-west-2:111122223333:log-
group:SecurityHubAPIAccessLog/Prod"
  },
  "AutoDeploy": false,
  "LastDeploymentStatusMessage": "Message",
  "ApiGatewayManaged": true,

```

}

AwsAutoScalingAutoScalingGroup

The `AwsAutoScalingAutoScalingGroup` object provides details about an Amazon EC2 Auto Scaling automatic scaling group.

The following is an example `AwsAutoScalingAutoScalingGroup` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsAutoScalingAutoScalingGroup` attributes, see [AwsAutoScalingAutoScalingGroupDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsAutoScalingAutoScalingGroup": {
  "CreatedTime": "2017-10-17T14:47:11Z",
  "HealthCheckGracePeriod": 300,
  "HealthCheckType": "EC2",
  "LaunchConfigurationName": "mylaunchconf",
  "LoadBalancerNames": [],
  "LaunchTemplate": {
    "LaunchTemplateId": "string",
    "LaunchTemplateName": "string",
    "Version": "string"
  },
  "MixedInstancesPolicy": {
    "InstancesDistribution": {
      "OnDemandAllocationStrategy": "prioritized",
      "OnDemandBaseCapacity": number,
      "OnDemandPercentageAboveBaseCapacity": number,
      "SpotAllocationStrategy": "lowest-price",
      "SpotInstancePools": number,
      "SpotMaxPrice": "string"
    },
    "LaunchTemplate": {
      "LaunchTemplateSpecification": {
        "LaunchTemplateId": "string",
        "LaunchTemplateName": "string",
        "Version": "string"
      },
      "CapacityRebalance": true,
      "Overrides": [
        {
          "InstanceType": "string",
          "WeightedCapacity": "string"
        }
      ]
    }
  }
}
```

The following is a list of valid value examples for `AwsAutoScalingAutoScalingGroup` attributes:

- `HealthCheckType`

Valid values: EC2 | ELB

- `OnDemandAllocationStrategy`

Valid values: prioritized

- `SpotAllocationStrategy`

Valid values: lowest-price | capacity-optimized | capacity-optimized-prioritized

AwsAutoScalingLaunchConfiguration

The `AwsAutoScalingLaunchConfiguration` object provides details about an Amazon EC2 Auto Scaling launch configuration.

The following is an example `AwsAutoScalingLaunchConfiguration` finding in the AWS Security Finding Format (ASFF).

To view descriptions of `AwsAutoScalingLaunchConfiguration` attributes, see [AwsAutoScalingLaunchConfigurationDetails](#) in the *AWS Security Hub API Reference*.

Example

```
AwsAutoScalingLaunchConfiguration: {
  "LaunchConfigurationName": "newtest",
  "ImageId": "ami-058a3739b02263842",
  "KeyName": "55hundredinstance",
  "SecurityGroups": [ "sg-01fce87ad6e019725" ],
  "ClassicLinkVpcSecurityGroups": [],
  "UserData": "...Base64-Encoded user data..."
  "InstanceType": "a1.metal",
  "KernelId": "",
  "RamdiskId": "ari-a51cf9cc",
  "BlockDeviceMappings": [
    {
      "DeviceName": "/dev/sdh",
      "Ebs": {
        "VolumeSize": 30,
        "VolumeType": "gp2",
        "DeleteOnTermination": false,
        "Encrypted": true,
        "SnapshotId": "snap-ffaale69",
        "VirtualName": "ephemeral1"
      }
    },
    {
      "DeviceName": "/dev/sdb",
      "NoDevice": true
    },
    {
      "DeviceName": "/dev/sda1",
      "Ebs": {
        "SnapshotId": "snap-02420cd3d2dea1bc0",
        "VolumeSize": 8,
        "VolumeType": "gp2",
        "DeleteOnTermination": true,
        "Encrypted": false
      }
    },
    {
      "DeviceName": "/dev/sdi",
      "Ebs": {
        "VolumeSize": 20,
        "VolumeType": "gp2",
        "DeleteOnTermination": false,
        "Encrypted": true
      }
    },
    {
      "DeviceName": "/dev/sdc",
      "NoDevice": true
    }
  ],
  "InstanceMonitoring": {
```

```

    "Enabled": false
  },
  "CreatedTime": 1620842933453,
  "EbsOptimized": false,
  "AssociatePublicIpAddress": true,
  "SpotPrice": "0.045"
}

```

The following is a list of valid value examples for `AwsAutoScalingLaunchConfiguration` attributes:

- `VolumeType`

Valid values: `standard` | `io1` | `gp2` | `st1` | `sc1` | `gp3`

AwsCertificateManagerCertificate

The `AwsCertificateManagerCertificate` object provides details about an AWS Certificate Manager (ACM) certificate.

The following is an example `AwsCertificateManagerCertificate` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCertificateManagerCertificate` attributes, see [AwsCertificateManagerCertificateDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsCertificateManagerCertificate": {
  "CertificateAuthorityArn": "arn:aws:acm:us-west-2:444455556666:certificate-authority/example",
  "CreatedAt": "2019-05-24T18:12:02.000Z",
  "DomainName": "example.amazondomains.com",
  "DomainValidationOptions": [
    {
      "DomainName": "example.amazondomains.com",
      "ResourceRecord": {
        "Name": "_1bacb61828d3a1020c40a560ceed08f7.example.amazondomains.com",
        "Type": "CNAME",
        "Value": "_example.acm-validations.aws."
      },
      "ValidationDomain": "example.amazondomains.com",
      "ValidationEmails": [],
      "ValidationMethod": "DNS",
      "ValidationStatus": "SUCCESS"
    }
  ],
  "ExtendedKeyUsages": [
    {
      "Name": "TLS_WEB_SERVER_AUTHENTICATION",
      "Oid": "1.3.6.1.5.5.7.3.1"
    },
    {
      "Name": "TLS_WEB_CLIENT_AUTHENTICATION",
      "Oid": "1.3.6.1.5.5.7.3.2"
    }
  ],
  "FailureReason": "",
  "ImportedAt": "2018-08-17T00:13:00.000Z",
  "InUseBy": ["arn:aws:amazondomains:us-west-2:444455556666:loadbalancer/example"],
  "IssuedAt": "2020-04-26T00:41:17.000Z",
  "Issuer": "Amazon",
  "KeyAlgorithm": "RSA-1024",
  "KeyUsages": [
    {
      "Name": "DIGITAL_SIGNATURE",

```

```

    },
    {
      "Name": "KEY_ENCIPHERMENT",
    }
  ],
  "NotAfter": "2021-05-26T12:00:00.000Z",
  "NotBefore": "2020-04-26T00:00:00.000Z",
  "Options": {
    "CertificateTransparencyLoggingPreference": "ENABLED",
  }
  "RenewalEligibility": "ELIGIBLE",
  "RenewalSummary": {
    "DomainValidationOptions": [
      {
        "DomainName": "example.amazondomains.com",
        "ResourceRecord": {
          "Name": "_lbacb61828d3a1020c40a560ceed08f7.example.amazondomains.com",
          "Type": "CNAME",
          "Value": "_example.acm-validations.aws.com",
        },
        "ValidationDomain": "example.amazondomains.com",
        "ValidationEmails": [],
        "ValidationMethod": "DNS",
        "ValidationStatus": "SUCCESS"
      }
    ],
    "RenewalStatus": "SUCCESS",
    "RenewalStatusReason": "",
    "UpdatedAt": "2020-04-26T00:41:35.000Z",
  },
  "Serial": "02:ac:86:b6:07:2f:0a:61:0e:3a:ac:fd:d9:ab:17:1a",
  "SignatureAlgorithm": "SHA256WITHRSA",
  "Status": "ISSUED",
  "Subject": "CN=example.amazondomains.com",
  "SubjectAlternativeNames": ["example.amazondomains.com"],
  "Type": "AMAZON_ISSUED"
}

```

The following is a list of valid value examples for `AwsCertificateManagerCertificate` attributes:

- `FailureReason`

Valid values: NO_AVAILABLE_CONTACTS | ADDITIONAL_VERIFICATION_REQUIRED | DOMAIN_NOT_ALLOWED | INVALID_PUBLIC_DOMAIN | DOMAIN_VALIDATION_DENIED | CAA_ERROR | PCA_LIMIT_EXCEEDED | PCA_INVALID_ARN | PCA_INVALID_STATE | PCA_REQUEST_FAILED | PCA_NAME_CONSTRAINTS_VALIDATION | PCA_RESOURCE_NOT_FOUND | PCA_INVALID_ARGS | PCA_INVALID_DURATION | PCA_ACCESS_DENIED | SLR_NOT_FOUND | OTHER

- `KeyAlgorithm`

RSA_2048 | RSA_1024 | RSA_4096 | EC_prime256v1 | EC_secp384r1 | EC_secp521r1

- `Status`

PENDING_VALIDATION | ISSUED | INACTIVE | EXPIRED | VALIDATION_TIMED_OUT | REVOKED | FAILED

- `Type`

IMPORTED | AMAZON_ISSUED | PRIVATE

- `RenewalStatus`

PENDING_AUTO_RENEWAL | PENDING_VALIDATION | SUCCESS | FAILED

- `RenewalStatusReason`

Valid values: NO_AVAILABLE_CONTACTS | ADDITIONAL_VERIFICATION_REQUIRED | DOMAIN_NOT_ALLOWED | INVALID_PUBLIC_DOMAIN | DOMAIN_VALIDATION_DENIED | CAA_ERROR | PCA_LIMIT_EXCEEDED | PCA_INVALID_ARN | PCA_INVALID_STATE | PCA_REQUEST_FAILED | PCA_NAME_CONSTRAINTS_VALIDATION | PCA_RESOURCE_NOT_FOUND | PCA_INVALID_ARGS | PCA_INVALID_DURATION | PCA_ACCESS_DENIED | SLR_NOT_FOUND | OTHER

AwsCloudFormationStack

The `AwsCloudFormationStack` object provides details about an AWS CloudFormation stack that is nested as a resource in a top-level template.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsCloudFormationStack` object. To view descriptions of `AwsCloudFormationStack` attributes, see [AwsCloudFormationStackDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudFormationStack": {
  "Capabilities": [
    "CAPABILITY_IAM",
    "CAPABILITY_NAMED_IAM"
  ],
  "CreationTime": "2022-02-18T15:31:53.161Z",
  "Description": "AWS CloudFormation Sample",
  "DisableRollback": true,
  "DriftInformation": {
    "StackDriftStatus": "DRIFTED"
  },
  "EnableTerminationProtection": false,
  "LastUpdatedTime": "2022-02-18T15:31:53.161Z",
  "NotificationArns": [
    "arn:aws:sns:us-east-1:978084797471:sample-sns-cfn"
  ],
  "Outputs": [{
    "Description": "URL for newly created LAMP stack",
    "OutputKey": "WebsiteUrl",
    "OutputValue": "http://ec2-44-193-18-241.compute-1.amazonaws.com"
  }],
  "RoleArn": "arn:aws:iam::012345678910:role/exampleRole",
  "StackId": "arn:aws:cloudformation:us-east-1:978084797471:stack/sample-stack/e5d9f7e0-90cf-11ec-88c6-12ac1f91724b",
  "StackName": "sample-stack",
  "StackStatus": "CREATE_COMPLETE",
  "StackStatusReason": "Success",
  "TimeoutInMinutes": 1
}
```

AwsCloudFrontDistribution

The `AwsCloudFrontDistribution` object provides details about a Amazon CloudFront distribution configuration.

The following is an example `AwsCloudFrontDistribution` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCloudFrontDistribution` attributes, see [AwsCloudFrontDistributionDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudFrontDistribution": {
  "CacheBehaviors": {
```

```

        "Items": [
            {
                "ViewerProtocolPolicy": "https-only"
            }
        ]
    },
    "DefaultCacheBehavior": {
        "ViewerProtocolPolicy": "https-only"
    },
    "DefaultRootObject": "index.html",
    "DomainName": "d2wkuj2w9l34gt.cloudfront.net",
    "Etag": "E37HOT42DHPVYH",
    "LastModifiedTime": "2015-08-31T21:11:29.093Z",
    "Logging": {
        "Bucket": "myawslogbucket.s3.amazonaws.com",
        "Enabled": false,
        "IncludeCookies": false,
        "Prefix": "myawslog/"
    },
    "OriginGroups": {
        "Items": [
            {
                "FailoverCriteria": {
                    "StatusCodes": {
                        "Items": [
                            200,
                            301,
                            404
                        ]
                    }
                }
            }
        ]
    },
    "Origins": {
        "Items": [
            {
                "CustomOriginConfig": {
                    "HttpPort": 80,
                    "HttpsPort": 443,
                    "OriginKeepaliveTimeout": 60,
                    "OriginProtocolPolicy": "match-viewer",
                    "OriginReadTimeout": 30,
                    "OriginSslProtocols": {
                        "Items": ["SSLv3", "TLSv1"],
                        "Quantity": 2
                    }
                }
            }
        ],
        "DomainName": "my-bucket.s3.amazonaws.com",
        "Id": "my-origin",
        "OriginPath": "/production",
        "S3OriginConfig": {
            "OriginAccessIdentity": "origin-access-identity/cloudfront/
E2YFS67H6VB6E4"
        }
    },
    "Status": "Deployed",
    "ViewerCertificate": {
        "AcmCertificateArn": "arn:aws:acm::123456789012:AcmCertificateArn",
        "Certificate": "ASCAJRRE5XYF52TKRY5M4",
        "CertificateSource": "iam",
    }
}

```



```
        "CloudFrontDefaultCertificate": true,  
        "IamCertificateId": "ASCAJRRE5XYF52TKRY5M4",  
        "MinimumProtocolVersion": "TLSv1.2_2021",  
        "SslSupportMethod": "sni-only"  
    },  
    "WebAclId": "waf-1234567890"  
}
```

AwsCloudTrailTrail

The `AwsCloudTrailTrail` object provides details about a AWS CloudTrail trail.

The following is an example `AwsCloudTrailTrail` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCloudTrailTrail` attributes, see [AwsCloudTrailTrailDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudTrailTrail": {  
  "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-west-2:123456789012:log-group:CloudTrail/  
regression:*",  
  "CloudWatchLogsRoleArn": "arn:aws:iam::866482105055:role/CloudTrail_CloudWatchLogs",  
  "HasCustomEventSelectors": true,  
  "HomeRegion": "us-west-2",  
  "IncludeGlobalServiceEvents": true,  
  "IsMultiRegionTrail": true,  
  "IsOrganizationTrail": false,  
  "KmsKeyId": "kmsKeyId",  
  "LogFileValidationEnabled": true,  
  "Name": "regression-trail",  
  "S3BucketName": "cloudtrail-bucket",  
  "S3KeyPrefix": "s3KeyPrefix",  
  "SnsTopicArn": "arn:aws:sns:us-east-2:123456789012:MyTopic",  
  "SnsTopicName": "snsTopicName",  
  "TrailArn": "arn:aws:cloudtrail:us-west-2:123456789012:trail"  
}
```

AwsCloudWatchAlarm

The `AwsCloudWatchAlarm` object provides details about Amazon CloudWatch alarms that watch a metric or perform an action when an alarm changes state.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsCloudWatchAlarm` object. To view descriptions of `AwsCloudWatchAlarm` attributes, see [AwsCloudWatchAlarmDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsCloudWatchAlarm": {  
  "ActonsEnabled": true,  
  "AlarmActions": [  
    "arn:aws:automate:region:ec2:stop",  
    "arn:aws:automate:region:ec2:terminate"  
  ],  
  "AlarmArn": "arn:aws:cloudwatch:us-west-2:012345678910:alarm:sampleAlarm",  
  "AlarmConfigurationUpdatedTimestamp": "2022-02-18T15:31:53.161Z",  
  "AlarmDescription": "Alarm Example",  
  "AlarmName": "Example",  
  "ComparisonOperator": "GreaterThanOrEqualToThreshold",  
  "DatapointsToAlarm": 1,  
  "Dimensions": [{  
    "Name": "InstanceId",  
  }]
```

```

    "Value": "i-1234567890abcdef0"
  }],
  "EvaluateLowSampleCountPercentile": "evaluate",
  "EvaluationPeriods": 1,
  "ExtendedStatistic": "p99.9",
  "InsufficientDataActions": [
    "arn:aws:automate:region:ec2:stop"
  ],
  "MetricName": "Sample Metric",
  "Namespace": "YourNamespace",
  "OkActions": [
    "arn:aws:swf:region:account-id:action/actions/AWS_EC2.InstanceId.Stop/1.0"
  ],
  "Period": 1,
  "Statistic": "SampleCount",
  "Threshold": 12.3,
  "ThresholdMetricId": "t1",
  "TreatMissingData": "notBreaching",
  "Unit": "Kilobytes/Second"
}

```

AwsCodeBuildProject

The `AwsCodeBuildProject` object provides information about an AWS CodeBuild project.

The following is an example `AwsCodeBuildProject` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsCodeBuildProject` attributes, see [AwsCodeBuildProjectDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsCodeBuildProject": {
  "Artifacts": [
    {
      "ArtifactIdentifier": "string",
      "EncryptionDisabled": boolean,
      "Location": "string",
      "Name": "string",
      "NamespaceType": "string",
      "OverrideArtifactName": boolean,
      "Packaging": "string",
      "Path": "string",
      "Type": "string"
    }
  ],
  "SecondaryArtifacts": [
    {
      "ArtifactIdentifier": "string",
      "EncryptionDisabled": boolean,
      "Location": "string",
      "Name": "string",
      "NamespaceType": "string",
      "OverrideArtifactName": boolean,
      "Packaging": "string",
      "Path": "string",
      "Type": "string"
    }
  ],
  "EncryptionKey": "string",
  "Certificate": "string",
  "Environment": {
    "Certificate": "string",
    "EnvironmentVariables": [
      {

```

```

        "Name": "string",
        "Type": "string",
        "Value": "string"
    }
],
"ImagePullCredentialsType": "string",
"PrivilegedMode": boolean,
"RegistryCredential": {
    "Credential": "string",
    "CredentialProvider": "string"
},
"Type": "string"
},
"LogsConfig": {
    "CloudWatchLogs": {
        "GroupName": "string",
        "Status": "string",
        "StreamName": "string"
    },
    "S3Logs": {
        "EncryptionDisabled": boolean,
        "Location": "string",
        "Status": "string"
    }
},
"Name": "string",
"ServiceRole": "string",
"Source": {
    "Type": "string",
    "Location": "string",
    "GitCloneDepth": integer
},
"VpcConfig": {
    "VpcId": "string",
    "Subnets": ["string"],
    "SecurityGroupIds": ["string"]
}
}

```

AwsDynamoDbTable

The `AwsDynamoDbTable` object provides details about an Amazon DynamoDB table.

The following is an example `AwsDynamoDbTable` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsDynamoDbTable` attributes, see [AwsDynamoDbTableDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsDynamoDbTable": {
    "AttributeDefinitions": [
        {
            "AttributeName": "attribute1",
            "AttributeType": "value 1"
        },
        {
            "AttributeName": "attribute2",
            "AttributeType": "value 2"
        },
        {
            "AttributeName": "attribute3",
            "AttributeType": "value 3"
        }
    ],
}

```

```

"BillingModeSummary": {
  "BillingMode": "PAY_PER_REQUEST",
  "LastUpdateToPayPerRequestDateTime": "2019-12-03T15:23:10.323Z"
},
"CreationDateTime": "2019-12-03T15:23:10.248Z",
"GlobalSecondaryIndexes": [
  {
    "Backfilling": false,
    "IndexArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable/index/
exampleIndex",
    "IndexName": "standardsControlArnIndex",
    "IndexSizeBytes": 1862513,
    "IndexStatus": "ACTIVE",
    "ItemCount": 20,
    "KeySchema": [
      {
        "AttributeName": "City",
        "KeyType": "HASH"
      },
      {
        "AttributeName": "Date",
        "KeyType": "RANGE"
      }
    ],
    "Projection": {
      "NonKeyAttributes": ["predictorName"],
      "ProjectionType": "ALL"
    },
    "ProvisionedThroughput": {
      "LastIncreaseDateTime": "2019-03-14T13:21:00.399Z",
      "LastDecreaseDateTime": "2019-03-14T12:47:35.193Z",
      "NumberOfDecreasesToday": 0,
      "ReadCapacityUnits": 100,
      "WriteCapacityUnits": 50
    }
  }
],
"GlobalTableVersion": "V1",
"ItemCount": 2705,
"KeySchema": [
  {
    "AttributeName": "zipcode",
    "KeyType": "HASH"
  }
],
"LatestStreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable/
stream/2019-12-03T23:23:10.248",
"LatestStreamLabel": "2019-12-03T23:23:10.248",
"LocalSecondaryIndexes": [
  {
    "IndexArn": "arn:aws:dynamodb:us-east-1:111122223333:table/exampleGroup/index/
exampleId",
    "IndexName": "CITY_DATE_INDEX_NAME",
    "KeySchema": [
      {
        "AttributeName": "zipcode",
        "KeyType": "HASH"
      }
    ],
    "Projection": {
      "NonKeyAttributes": ["predictorName"],
      "ProjectionType": "ALL"
    }
  }
],
"ProvisionedThroughput": {

```

```

        "LastIncreaseDateTime": "2019-03-14T13:21:00.399Z",
        "LastDecreaseDateTime": "2019-03-14T12:47:35.193Z",
        "NumberOfDecreasesToday": 0,
        "ReadCapacityUnits": 100,
        "WriteCapacityUnits": 50
    },
    "Replicas": [
        {
            "GlobalSecondaryIndexes": [
                {
                    "IndexName": "CITY_DATE_INDEX_NAME",
                    "ProvisionedThroughputOverride": {
                        "ReadCapacityUnits": 10
                    }
                }
            ],
            "KmsMasterKeyId": "KmsKeyId",
            "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": 10
            },
            "RegionName": "regionName",
            "ReplicaStatus": "CREATING",
            "ReplicaStatusDescription": "replicaStatusDescription"
        }
    ],
    "RestoreSummary": {
        "SourceBackupArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable/backup/backup1",
        "SourceTableArn": "arn:aws:dynamodb:us-west-2:111122223333:table/exampleTable",
        "RestoreDateTime": "2020-06-22T17:40:12.322Z",
        "RestoreInProgress": true
    },
    "SseDescription": {
        "InaccessibleEncryptionDateTime": "2018-01-26T23:50:05.000Z",
        "Status": "ENABLED",
        "SseType": "KMS",
        "KmsMasterKeyArn": "arn:aws:kms:us-east-1:111122223333:key/key1"
    },
    "StreamSpecification": {
        "StreamEnabled": true,
        "StreamViewType": "NEW_IMAGE"
    },
    "TableId": "example-table-id-1",
    "TableName": "example-table",
    "TableSizeBytes": 1862513,
    "TableStatus": "ACTIVE"
}

```

The following is a list of valid value examples for `AwsDynamoDbTable` attributes:

- `TableStatus`

CREATING | UPDATING | DELETING | ACTIVE | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED

- `IndexStatus`

CREATING | UPDATING | DELETING | ACTIVE

- `KeyType`

HASH | RANGE

- `ProjectionType`

ALL | KEYS_ONLY | INCLUDE

- ReplicaStatus

CREATING | CREATION_FAILED | UPDATING | DELETING | ACTIVE

AwsEc2Eip

The `AwsEc2Eip` object provides information about an Elastic IP address.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Eip` object. To view descriptions of `AwsEc2Eip` attributes, see [AwsEc2EipDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2Eip": {
  "InstanceId": "instance1",
  "PublicIp": "192.0.2.04",
  "AllocationId": "eipalloc-example-id-1",
  "AssociationId": "eipassoc-example-id-1",
  "Domain": "vpc",
  "PublicIpv4Pool": "anycompany",
  "NetworkBorderGroup": "eu-central-1",
  "NetworkInterfaceId": "eni-example-id-1",
  "NetworkInterfaceOwnerId": "777788889999",
  "PrivateIpAddress": "192.0.2.03"
}
```

AwsEc2Instance

The `AwsEc2Instance` object provides details about an Amazon EC2 instance.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Instance` object. To view descriptions of `AwsEc2Instance` attributes, see [AwsEc2InstanceDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2Instance": {
  "IamInstanceProfileArn": "arn:aws:iam::123456789012:role/IamInstanceProfileArn",
  "ImageId": "ami-79fd7eee",
  "IPv4Addresses": ["1.1.1.1"],
  "IPv6Addresses": ["2001:db8:1234:1a2b::123"],
  "KeyName": "testkey",
  "LaunchedAt": "2018-09-29T01:25:54Z",
  "MetadataOptions": {
    "HttpEndpoint": "enabled",
    "HttpProtocolIpv6": "enabled",
    "HttpPutResponseHopLimit": 1,
    "HttpTokens": "optional",
    "InstanceMetadataTags": "disabled"
  },
  "NetworkInterfaces": [
    {
      "NetworkInterfaceId": "eni-e5aa89a3"
    }
  ],
  "SubnetId": "PublicSubnet",
  "Type": "i3.xlarge",
  "VirtualizationType": "hvm",
  "VpcId": "TestVPCIPv6"
}
```

AwsEc2NetworkAcl

The `AwsEc2NetworkAcl` object contains details about an Amazon EC2 network access control list (ACL).

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2NetworkAcl` object. To view descriptions of `AwsEc2NetworkAcl` attributes, see [AwsEc2NetworkAclDetails](#) in the *AWS Security Hub API Reference*.

Example

```
AwsEc2NetworkAcl: {
  "IsDefault": false,
  "NetworkAclId": "acl-1234567890abcdef0",
  "OwnerId": "123456789012",
  "VpcId": "vpc-1234abcd",
  "Associations": [{
    "NetworkAclAssociationId": "aclassoc-abcd1234",
    "NetworkAclId": "acl-021345abcdef6789",
    "SubnetId": "subnet-abcd1234"
  }],
  "Entries": [{
    "CidrBlock": "10.24.34.0/23",
    "Egress": true,
    "IcmpTypeCode": {
      "Code": 10,
      "Type": 30
    },
    "Ipv6CidrBlock": "2001:DB8::/32",
    "PortRange": {
      "From": 20,
      "To": 40
    },
    "Protocol": "tcp",
    "RuleAction": "allow",
    "RuleNumber": 100
  }]
}
```

AwsEc2NetworkInterface

The `AwsEc2NetworkInterface` object provides information about an Amazon EC2 network interface.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2NetworkInterface` object. To view descriptions of `AwsEc2NetworkInterface` attributes, see [AwsEc2NetworkInterfaceDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2NetworkInterface": {
  "Attachment": {
    "AttachTime": "2019-01-01T03:03:21Z",
    "AttachmentId": "eni-attach-43348162",
    "DeleteOnTermination": true,
    "DeviceIndex": 123,
    "InstanceId": "i-1234567890abcdef0",
    "InstanceOwnerId": "123456789012",
    "Status": 'ATTACHED'
  },
  "SecurityGroups": [
    {
      "GroupName": "my-security-group",
      "GroupId": "sg-903004f8"
    }
  ]
}
```

```
  ],  
  "NetworkInterfaceId": 'eni-686ea200',  
  "SourceDestCheck": false  
}
```

AwsEc2SecurityGroup

The `AwsEc2SecurityGroup` object describes an Amazon EC2 security group.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2SecurityGroup` object. To view descriptions of `AwsEc2SecurityGroup` attributes, see [AwsEc2SecurityGroupDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2SecurityGroup": {  
  "GroupName": "MySecurityGroup",  
  "GroupId": "sg-903004f8",  
  "OwnerId": "123456789012",  
  "VpcId": "vpc-1a2b3c4d",  
  "IpPermissions": [  
    {  
      "IpProtocol": "-1",  
      "IpRanges": [],  
      "UserIdGroupPairs": [  
        {  
          "UserId": "123456789012",  
          "GroupId": "sg-903004f8"  
        }  
      ],  
      "PrefixListIds": [  
        {"PrefixListId": "pl-63a5400a"}  
      ]  
    },  
    {  
      "PrefixListIds": [],  
      "FromPort": 22,  
      "IpRanges": [  
        {  
          "CidrIp": "203.0.113.0/24"  
        }  
      ],  
      "ToPort": 22,  
      "IpProtocol": "tcp",  
      "UserIdGroupPairs": []  
    }  
  ]  
}
```

AwsEc2Subnet

The `AwsEc2Subnet` object provides information about a subnet in Amazon EC2.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Subnet` object. To view descriptions of `AwsEc2Subnet` attributes, see [AwsEc2SubnetDetails](#) in the *AWS Security Hub API Reference*.

Example

```
AwsEc2Subnet: {  
  "AssignIpv6AddressOnCreation": false,  
  "AvailabilityZone": "us-west-2c",
```



```
"AvailabilityZoneId": "usw2-az3",
"AvailableIpAddressCount": 8185,
"CidrBlock": "10.0.0.0/24",
"DefaultForAz": false,
"MapPublicIpOnLaunch": false,
"OwnerId": "123456789012",
"State": "available",
"SubnetArn": "arn:aws:ec2:us-west-2:123456789012:subnet/subnet-d5436c93",
"SubnetId": "subnet-d5436c93",
"VpcId": "vpc-153ade70",
"Ipv6CidrBlockAssociationSet": [{
  "AssociationId": "subnet-cidr-assoc-EXAMPLE",
  "Ipv6CidrBlock": "2001:DB8::/32",
  "CidrBlockState": "associated"
}]
}
```

The following is a list of valid value examples for `AwsEc2Subnet` attributes:

- `State`

Valid values: `pending` | `available`

- `CidrBlockState`

Valid values: `associating` | `associated` | `disassociating` | `disassociated` | `failing` | `failed`

AwsEc2TransitGateway

The `AwsEc2TransitGateway` object provides details about virtual private clouds (VPCs) and on-premises networks.

The following is an example `AwsEc2TransitGateway` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsEc2TransitGateway` attributes, see [AwsEc2TransitGatewayDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2TransitGateway": {
  "AmazonSideAsn": 65000,
  "AssociationDefaultRouteTableId": "tgw-rtb-099ba47cbbea837cc",
  "AutoAcceptSharedAttachments": "disable",
  "DefaultRouteTableAssociation": "enable",
  "DefaultRouteTablePropagation": "enable",
  "Description": "sample transit gateway",
  "DnsSupport": "enable",
  "Id": "tgw-042ae6bf7a5c126c3",
  "MulticastSupport": "disable",
  "PropagationDefaultRouteTableId": "tgw-rtb-099ba47cbbea837cc",
  "TransitGatewayCidrBlocks": ["10.0.0.0/16"],
  "VpnEcmpSupport": "enable"
}
```

AwsEc2Volume

The `AwsEc2Volume` object provides details about an Amazon Elastic Compute Cloud volume.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Volume` object. To view descriptions of `AwsEc2Volume` attributes, see [AwsEc2VolumeDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsEc2Volume": {
  "Attachments": [{
    "AttachTime": "2017-10-17T14:47:11Z",
    "DeleteOnTermination": true,
    "InstanceId": "i-123abc456def789g",
    "Status": "attached"
  }],
  "CreateTime": "2020-02-24T15:54:30Z",
  "DeviceName": "/dev/sda2",
  "Encrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:111122223333:key/wJalrXUtnFEMI/K7MDENG/
bPxRfiCYEXAMPLEKEY",
  "Size": 80,
  "SnapshotId": "snap-014f23111111fadfb",
  "Status": "available",
  "VolumeId": "vol-11111111aae112f",
  "VolumeScanStatus": "scanned",
  "VolumeType": "gp2"
}

```

The following is a list of valid value examples for `AwsEc2Volume` attributes:

- Attachments Status

Valid values: attaching | attached | detaching | detached | busy

- Status

Valid values: creating | available | in-use | deleting | deleted | error

- VolumeScanStatus

Valid values: scanned | skipped

AwsEc2Vpc

The `AwsEc2Vpc` object provides details about an Amazon Elastic Compute Cloud virtual private cloud (VPC).

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2Vpc` object. To view descriptions of `AwsEc2Vpc` attributes, see [AwsEc2VpcDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsEc2Vpc": {
  "CidrBlockAssociationSet": [
    {
      "AssociationId": "vpc-cidr-assoc-0dc4c852f52abda97",
      "CidrBlock": "192.0.2.0/24",
      "CidrBlockState": "associated"
    }
  ],
  "DhcpOptionsId": "dopt-4e42ce28",
  "Ipv6CidrBlockAssociationSet": [
    {
      "AssociationId": "vpc-cidr-assoc-0dc4c852f52abda97",
      "CidrBlockState": "associated",
      "Ipv6CidrBlock": "192.0.2.0/24"
    }
  ],
  "State": "available"
}

```

```
}

```

The following is a list of valid value examples for `AwsEc2Vpc` attributes:

- `State`

Valid values: `pending` | `available`

AwsEc2VpcEndpointService

The `AwsEc2VpcEndpointService` object contains details about the service configuration for a VPC endpoint service.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2VpcEndpointService` object. To view descriptions of `AwsEc2VpcEndpointService` attributes, see [AwsEc2VpcEndpointServiceDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEc2VpcEndpointService": {
  "ServiceType": [
    {
      "ServiceType": "Interface"
    }
  ],
  "ServiceId": "vpce-svc-example1",
  "ServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example1",
  "ServiceState": "Available",
  "AvailabilityZones": [
    "us-east-1"
  ],
  "AcceptanceRequired": true,
  "ManagesVpcEndpoints": false,
  "NetworkLoadBalancerArns": [
    "arn:aws:elasticloadbalancing:us-east-1:444455556666:loadbalancer/net/my-network-load-balancer/example1"
  ],
  "GatewayLoadBalancerArns": [],
  "BaseEndpointDnsNames": [
    "vpce-svc-04eec859668b51c34.us-east-1.vpce.amazonaws.com"
  ],
  "PrivateDnsName": "my-private-dns"
}
```

The following is a list of valid value examples for `AwsEc2VpcEndpointService` attributes:

- `ServiceState`

Valid values: `Pending` | `Available` | `Deleting` | `Deleted` | `Failed`

AwsEc2VpcPeeringConnection

The `AwsCloudFormationStack` object provides details about the networking connection between two VPCs.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2VpcPeeringConnection` object. To view descriptions of `AwsEc2VpcPeeringConnection` attributes, see [AwsEc2VpcPeeringConnectionDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsEc2VpcPeeringConnection": {
  "AcceptorVpcInfo": {
    "CidrBlock": "10.0.0.0/28",
    "CidrBlockSet": [{
      "CidrBlock": "10.0.0.0/28"
    }],
    "Ipv6CidrBlockSet": [{
      "Ipv6CidrBlock": "2002::1234:abcd:ffff:c0a8:101/64"
    }],
    "OwnerId": "012345678910",
    "PeeringOptions": {
      "AllowDnsResolutionFromRemoteVpc": true,
      "AllowEgressFromLocalClassicLinkToRemoteVpc": false,
      "AllowEgressFromLocalVpcToRemoteClassicLink": true
    },
    "Region": "us-west-2",
    "VpcId": "vpc-i123456"
  },
  "ExpirationTime": "2022-02-18T15:31:53.161Z",
  "RequesterVpcInfo": {
    "CidrBlock": "192.168.0.0/28",
    "CidrBlockSet": [{
      "CidrBlock": "192.168.0.0/28"
    }],
    "Ipv6CidrBlockSet": [{
      "Ipv6CidrBlock": "2002::1234:abcd:ffff:c0a8:101/64"
    }],
    "OwnerId": "012345678910",
    "PeeringOptions": {
      "AllowDnsResolutionFromRemoteVpc": true,
      "AllowEgressFromLocalClassicLinkToRemoteVpc": false,
      "AllowEgressFromLocalVpcToRemoteClassicLink": true
    },
    "Region": "us-west-2",
    "VpcId": "vpc-i123456"
  },
  "Status": {
    "Code": "initiating-request",
    "Message": "Active"
  },
  "VpcPeeringConnectionId": "pcx-1a2b3c4d"
}

```

AwsEc2VpnConnection

The `AwsEc2VpnConnection` object provides details about an Amazon EC2 VPN connection.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEc2VpnConnection` object. To view descriptions of `AwsEc2VpnConnection` attributes, see [AwsEc2VpnConnectionDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsEc2VpnConnection": {
  "VpnConnectionId": "vpn-205e4f41",
  "State": "available",
  "CustomerGatewayConfiguration": "",
  "CustomerGatewayId": "cgw-5699703f",
  "Type": "ipsec.1",
  "VpnGatewayId": "vgw-2ccb2245",
  "Category": "VPN",
  "TransitGatewayId": "tgw-09b6f3a659e2b5elf",
  "VgwTelemetry": [

```

```

    {
      "OutsideIpAddress": "92.0.2.11",
      "Status": "DOWN",
      "LastStatusChange": "2016-11-11T23:09:32.000Z",
      "StatusMessage": "IPSEC IS DOWN",
      "AcceptedRouteCount": 0
    },
    {
      "OutsideIpAddress": "92.0.2.12",
      "Status": "DOWN",
      "LastStatusChange": "2016-11-11T23:10:51.000Z",
      "StatusMessage": "IPSEC IS DOWN",
      "AcceptedRouteCount": 0
    }
  ],
  "Routes": [{
    "DestinationCidrBlock": "10.24.34.0/24",
    "State": "available"
  }],
  "Options": {
    "StaticRoutesOnly": true
    "TunnelOptions": [{
      "DpdTimeoutSeconds": 30,
      "IkeVersions": ["ikev1", "ikev2"],
      "Phase1DhGroupNumbers": [14, 15, 16, 17, 18],
      "Phase1EncryptionAlgorithms": ["AES128", "AES256"],
      "Phase1IntegrityAlgorithms": ["SHA1", "SHA2-256"],
      "Phase1LifetimeSeconds": 28800,
      "Phase2DhGroupNumbers": [14, 15, 16, 17, 18],
      "Phase2EncryptionAlgorithms": ["AES128", "AES256"],
      "Phase2IntegrityAlgorithms": ["SHA1", "SHA2-256"],
      "Phase2LifetimeSeconds": 28800,
      "PreSharedKey": "RltXC3REhTw1RAdim2s1uMfkkSDLyGJoe1QEWGxqkQ=",
      "RekeyFuzzPercentage": 100,
      "RekeyMarginTimeSeconds": 540,
      "ReplayWindowSize": 1024,
      "TunnelInsideCidr": "10.24.34.0/23"
    }
  ]
}

```

The following is a list of valid value examples for `AwsCertificateManagerCertificate` attributes:

- `State`
Valid values: `pendingng` | `available` | `deleting` | `deleted`
- `VgwTelemetry Status`
Valid values: `UP` | `DOWN`

AwsEcrContainerImage

The `AwsEcrContainerImage` object provides information about an Amazon Elastic Container Registry image.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcrContainerImage` object. To view descriptions of `AwsEcrContainerImage` attributes, see [AwsEcrContainerImageDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEcrContainerImage": {
```

```
"RegistryId": "123456789012",
"RepositoryName": "repository-name",
"Architecture": "amd64"
"ImageDigest":
"sha256:a568e5c7a953fbaaa2904ac83401f93e4a076972dc1bae527832f5349cd2fb10",
"ImageTags": ["00000000-0000-0000-0000-000000000000"],
"ImagePublishedAt": "2019-10-01T20:06:12Z"
}
```

The following is a list of valid value examples for `AwsEcrContainerImage` attributes:

- Architecture

Valid values: `i386` | `x86_64` | `arm64`

AwsEcrRepository

The `AwsEcrRepository` object provides information about an Amazon Elastic Container Registry repository.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcrRepository` object. To view descriptions of `AwsEcrRepository` attributes, see [AwsEcrRepositoryDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEcrRepository": {
  "LifecyclePolicy": {
    "RegistryId": "123456789012",
  },
  "RepositoryName": "sample-repo",
  "Arn": "arn:aws:ecr:us-west-2:111122223333:repository/sample-repo",
  "ImageScanningConfiguration": {
    "ScanOnPush": true
  },
  "ImageTagMutability": "IMMUTABLE"
}
```

The following is a list of valid value examples for `AwsEcrRepository` attributes:

- ImageTagMutability

Valid values: `MUTABLE` | `IMMUTABLE`

AwsEcsCluster

The `AwsEcsCluster` object provides details about an Amazon Elastic Container Service cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsCluster` object. To view descriptions of `AwsEcsCluster` attributes, see [AwsEcsClusterDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEcsCluster": {
  "ActiveServicesCount": 1,
  "CapacityProviders": [{
    "Type": "AWS::ECS::CapacityProvider",
    "Properties": {
```

```

        "AutoScalingGroupProvider": {
            "AutoScalingGroupArn": "arn:aws:autoscaling:us-
west-2:123456789012:autoScalingGroup:alb2c3d4-5678-90ab-cdef-
EXAMPLE11111:autoScalingGroupName/MyAutoScalingGroup",
            "ManagedScaling": {
                "MaximumScalingStepSize": 10,
                "MinimumScalingStepSize": 1,
                "Status": "ENABLED",
                "TargetCapacity": 100
            },
            "ManagedTerminationProtection": "ENABLED"
        },
        "Tags": [{
            "Key": "environment",
            "Value": "production"
        }
    ]
},
"ClusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/default",
"ClusterName": "default",
"ClusterSettings": [{
    "Name": "containerInsights",
    "Value": "enabled"
}],
"Configuration": {
    "ExecuteCommandConfiguration": {
        "KmsKeyId": "kmsKeyId",
        "LogConfiguration": {
            "CloudWatchEncryptionEnabled": true,
            "CloudWatchLogGroupName": "cloudWatchLogGroupName",
            "S3BucketName": "s3BucketName",
            "S3EncryptionEnabled": true,
            "S3KeyPrefix": "s3KeyPrefix"
        },
        "Logging": "DEFAULT"
    }
},
"DefaultCapacityProviderStrategy": [{
    "Base": 0,
    "CapacityProvider": "capacityProvider",
    "Weight": 1
}],
"RegisteredContainerInstancesCount": 1,
"RunningTasksCount": 1,
"Status": "ACTIVE"
}

```

The following is a list of valid value examples for `AwsCertificateManagerCertificate` attributes:

- `ClusterSettings Name`
Valid value: `containerInsights`
- `ClusterSettings Value`
Valid values: `enabled` | `disabled`

AwsEcsContainer

The `AwsEcsContainer` object contains details about an Amazon ECS container.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsContainer` object. To view descriptions of `AwsEcsContainer` attributes, see [AwsEcsContainerDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEcsContainer": {
  "Image": "11111111/
knotejs@sha256:356131c9fef111111111111115f4ed8de5f9dce4dc3bd34bg21846588a3",
  "MountPoints": [{
    "ContainerPath": "/mnt/etc",
    "SourceVolume": "vol-03909e9"
  }],
  "Name": "knote",
  "Privileged": true
}
```

AwsEcsService

The `AwsEcsService` object provides details about a service within an Amazon ECS cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsService` object. To view descriptions of `AwsEcsService` attributes, see [AwsEcsServiceDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEcsService": {
  "CapacityProviderStrategy": [
    {
      "Base": 12,
      "CapacityProvider": "",
      "Weight": ""
    }
  ],
  "Cluster": "arn:aws:ecs:us-east-1:111122223333:cluster/example-ecs-cluster",
  "DeploymentConfiguration": {
    "DeploymentCircuitBreaker": {
      "Enable": false,
      "Rollback": false
    },
    "MaximumPercent": 200,
    "MinimumHealthyPercent": 100
  },
  "DeploymentController": "",
  "DesiredCount": 1,
  "EnableEcsManagedTags": false,
  "EnableExecuteCommand": false,
  "HealthCheckGracePeriodSeconds": 1,
  "LaunchType": "FARGATE",
  "LoadBalancers": [
    {
      "ContainerName": "",
      "ContainerPort": 23,
      "LoadBalancerName": "",
      "TargetGroupArn": ""
    }
  ],
  "Name": "sample-app-service",
  "NetworkConfiguration": {
    "AwsVpcConfiguration": {
      "Subnets": [
        "Subnet-example1",
        "Subnet-example2"
      ],
      "SecurityGroups": [
        "Sg-0ce48e9a6e5b457f5"
      ]
    }
  }
}
```



```

    ],
    "AssignPublicIp": "ENABLED"
  }
},
"PlacementConstraints": [
  {
    "Expression": "",
    "Type": ""
  }
],
"PlacementStrategies": [
  {
    "Field": "",
    "Type": ""
  }
],
"PlatformVersion": "LATEST",
"PropagateTags": "",
"Role": "arn:aws:iam::111122223333:role/aws-servicerole/ecs.amazonaws.com/ServiceRoleForECS",
"SchedulingStrategy": "REPLICA",
"ServiceName": "sample-app-service",
"ServiceArn": "arn:aws:ecs:us-east-1:111122223333:service/example-ecs-cluster/sample-app-service",
"ServiceRegistries": [
  {
    "ContainerName": "",
    "ContainerPort": 1212,
    "Port": 1221,
    "RegistryArn": ""
  }
],
"TaskDefinition": "arn:aws:ecs:us-east-1:111122223333:task-definition/example-taskdef:1"
}

```

AwsEcsTask

Provides details about an Amazon Elastic Container Service task.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsTask` object. To view descriptions of `AwsEcsTask` attributes, see [AwsEcsTask](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsEcsTask": {
  "ClusterArn": "arn:aws:ecs:us-west-2:123456789012:task/MyCluster/1234567890123456789",
  "CreatedAt": "1557134011644",
  "Group": "service:fargate-service",
  "StartedAt": "1557134011644",
  "StartedBy": "ecs-svc/1234567890123456789",
  "TaskDefinitionArn": "arn:aws:ecs:us-west-2:123456789012:task-definition/sample-fargate:2",
  "Version": 3,
  "Volumes": [{
    "Name": "string",
    "Host": {
      "SourcePath": "string"
    }
  }
],
"Containers": {
  "Image": "1111111/knotejs@sha256:356131c9fef111111111111115f4ed8de5f9dce4dc3bd34bg21846588a3",

```

```

"MountPoints": [{
  "ContainerPath": "/mnt/etc",
  "SourceVolume": "vol-03909e9"
}],
"Name": "knote",
"Privileged": true
}
}

```

AwsEcsTaskDefinition

The `AwsEcsTaskDefinition` object contains details about a task definition. A task definition describes the container and volume definitions of an Amazon Elastic Container Service task.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEcsTaskDefinition` object. To view descriptions of `AwsEcsTaskDefinition` attributes, see [AwsEcsTaskDefinitionDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsEcsTaskDefinition": {
  "ContainerDefinitions": [
    {
      "Command": ['ruby', 'hi.rb'],
      "Cpu": 128,
      "Essential": true,
      "HealthCheck": {
        "Command": ["CMD-SHELL", "curl -f http://localhost/ || exit 1"],
        "Interval": 10,
        "Retries": 3,
        "StartPeriod": 5,
        "Timeout": 20
      },
      "Image": "tongueroo/sinatra:latest",
      "Interactive": true,
      "Links": [],
      "LogConfiguration": {
        "LogDriver": "awslogs",
        "Options": {
          "awslogs-group": "/ecs/sinatra-hi",
          "awslogs-region": "ap-southeast-1",
          "awslogs-stream-prefix": "ecs"
        },
        "SecretOptions": []
      },
      "MemoryReservation": 128,
      "Name": "web",
      "PortMappings": [
        {
          "ContainerPort": 4567,
          "HostPort": 4567,
          "Protocol": "tcp"
        }
      ],
      "Privileged": true,
      "StartTimeout": 10,
      "StopTimeout": 100,
    }
  ],
  "Family": "sinatra-hi",
  "NetworkMode": "host",
  "RequiresCompatibilities": ["EC2"],

```

```
    "TaskRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
  }
```

The following is a list of valid value examples for `AwsEcsTaskDefinition` attributes:

- `Cpu`

Valid values for the Fargate launch type: 256 (.25 vCPU) | 512 (.5 vCPU) | 1024 (1 vCPU) | 2048 (2 vCPU) | 4096 (4 vCPU)

- `IpMode`

Valid values: `host` | `task` | `none`

- `Memory`

For the EC2 launch type, can provide either a task-level memory value or a container-level memory value.

For the Fargate launch type, the available values for memory are based on the value of the `Cpu` attribute.

Cpu	Available values for Memory
256 (.25 vCPU)	512 (0.5 GB) 1024 (1 GB) 2048 (2 GB)
512 (.5 vCPU)	1024 (1 GB) 2048 (2 GB) 3072 (3 GB) 4096 (4 GB)
1024 (1 vCPU)	2048 (2 GB) 3072 (3 GB) 4096 (4 GB) 5120 (5 GB) 6144 (6 GB) 7168 (7 GB) 8192 (8 GB)
2048 (2 vCPU)	Between 4096 (4 GB) and 16384 (16 GB) in increments of 1024 (1 GB)
4096 (4 vCPU)	Between 8192 (8 GB) and 30720 (30 GB) in increments of 1024 (1 GB)

- `NetworkMode`

Valid values: `awsvpc` | `bridge` | `host` | `none`

- `PidMode`

Valid values: `host` | `task`

- `ContainerDefinitions DependsOn Condition`

Valid values: `START` | `COMPLETE` | `SUCCESS` | `HEALTHY`

- `ContainerDefinitions EnvironmentFiles Type`

Valid value: `s3`

- `ContainerDefinitions FirelensConfiguration Type`

Valid values: `fluentd` | `fluentbit`

- `ContainerDefinitions LinuxParameters Capabilities Add`

Valid values: `"ALL"` | `"AUDIT_CONTROL"` | `"AUDIT_WRITE"` | `"BLOCK_SUSPEND"` | `"CHOWN"` | `"DAC_OVERRIDE"` | `"DAC_READ_SEARCH"` | `"FOWNER"` | `"FSETID"` | `"IPC_LOCK"` | `"IPC_OWNER"` | `"KILL"` | `"LEASE"` | `"LINUX_IMMUTABLE"` | `"MAC_ADMIN"` | `"MAC_OVERRIDE"` | `"MKNOD"` | `"NET_ADMIN"` | `"NET_BIND_SERVICE"` | `"NET_BROADCAST"` | `"NET_RAW"` | `"SETFCAP"` | `"SETGID"`

```
| "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" |  
"SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" |  
"SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"
```

- ContainerDefinitions LinuxParameters Capabilities Drop

```
Valid values: "ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" |  
"DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" |  
"KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" |  
"NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" |  
"SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" |  
"SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" |  
"SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"
```

- ContainerDefinitions LinuxParameters Tmpfs MountOptions

```
Valid values: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" |  
"noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" |  
"noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" |  
"private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" |  
"norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" |  
"nr_blocks" | "mpol"
```

- ContainerDefinitions LogConfiguration LogDriver

Valid values on AWS Fargate: awslogs | splunk | awsfirelens

Valid values on EC2 instances: awslogs | fluentd | gelf | json-file | journald | logentries |
syslog | splunk | awsfirelens

- ContainerDefinitions ResourceRequirements Type

Valid values: GPU | InferenceAccelerator

- ContainerDefinitions Ulimits Name

Valid values: core | cpu | data | fsize | locks | memlock | msgqueue | nice | nofile | nproc | rss |
rtprio | rtttime | sigpending | stack

- Volumes DockerVolumeConfiguration Scope

Valid values: shared | task

AwsEfsAccessPoint

The `AwsEfsAccessPoint` object provides details about files stored in Amazon EFS.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEfsAccessPoint` object. To view descriptions of `AwsEfsAccessPoint` attributes, see [AwsEfsAccessPointDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsEfsAccessPoint": {  
  "AccessPointId": "fsap-05c4c0e79ba0b118a",  
  "Arn": "arn:aws:elasticfilesystem:us-east-1:863155670886:access-point/  
fsap-05c4c0e79ba0b118a",  
  "ClientToken": "AccessPointCompliant-ASK06ZZSxsEp",  
  "FileSystemId": "fs-0f8137f731cb32146",  
  "PosixUser": {  
    "Gid": "1000",  
    "SecondaryGids": ["0", "4294967295"],  
    "Uid": "1234"
```

```
},
"RootDirectory": {
  "CreationInfo": {
    "OwnerGid": "1000",
    "OwnerUid": "1234",
    "Permissions": "777"
  },
  "Path": "/tmp/example"
}
}
```

AwsEksCluster

The `AwsEksCluster` object provides details about an Amazon EKS cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsEksCluster` object. To view descriptions of `AwsEksCluster` attributes, see [AwsEksClusterDetails](#) in the *AWS Security Hub API Reference*.

Example

```
{
  "AwsEksCluster": {
    "Name": "example",
    "Arn": "arn:aws:eks:us-west-2:222222222222:cluster/example",
    "CreatedAt": 1565804921.901,
    "Version": "1.12",
    "RoleArn": "arn:aws:iam::222222222222:role/example-cluster-ServiceRole-1XWBQWYSFRE2Q",
    "ResourcesVpcConfig": {
      "SubnetIds": [
        "subnet-021345abcdef6789",
        "subnet-abcdef01234567890",
        "subnet-1234567890abcdef0"
      ],
      "SecurityGroupIds": [
        "sg-abcdef01234567890"
      ]
    },
    "Logging": {
      "ClusterLogging": [
        {
          "Types": [
            "api",
            "audit",
            "authenticator",
            "controllerManager",
            "scheduler"
          ],
          "Enabled": true
        }
      ]
    },
    "Status": "CREATING",
    "CertificateAuthorityData": {},
  }
}
```

The following is a list of valid value examples for `AwsEksCluster` attributes:

- `ClusterStatus`

Valid values: `CREATING` | `ACTIVE` | `DELETING` | `FAILED` | `UPDATING` | `PENDING`

- ClusterLogging Types

Valid values: api | audit | authenticator | controllerManager | scheduler

AwsElasticBeanstalkEnvironment

The `AwsElasticBeanstalkEnvironment` object contains details about an AWS Elastic Beanstalk environment.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElasticBeanstalkEnvironment` object. To view descriptions of `AwsElasticBeanstalkEnvironment` attributes, see [AwsElasticBeanstalkEnvironmentDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsElasticBeanstalkEnvironment": {
  "ApplicationName": "MyApplication",
  "Cname": "myexampleapp-env.devo-2.elasticbeanstalk-internal.com",
  "DateCreated": "2021-04-30T01:38:01.090Z",
  "DateUpdated": "2021-04-30T01:38:01.090Z",
  "Description": "Example description of my awesome application",
  "EndpointUrl": "eb-dv-e-p-AWSEBLoa-abcdef01234567890-021345abcdef6789.us-east-1.elb.amazonaws.com",
  "EnvironmentArn": "arn:aws:elasticbeanstalk:us-east-1:123456789012:environment/MyApplication/myapplication-env",
  "EnvironmentId": "e-abcd1234",
  "EnvironmentLinks": [
    {
      "EnvironmentName": "myexampleapp-env",
      "LinkName": "myapplicationLink"
    }
  ],
  "EnvironmentName": "myapplication-env",
  "OptionSettings": [
    {
      "Namespace": "aws:elasticbeanstalk:command",
      "OptionName": "BatchSize",
      "Value": "100"
    },
    {
      "Namespace": "aws:elasticbeanstalk:command",
      "OptionName": "Timeout",
      "Value": "600"
    },
    {
      "Namespace": "aws:elasticbeanstalk:command",
      "OptionName": "BatchSizeType",
      "Value": "Percentage"
    },
    {
      "Namespace": "aws:elasticbeanstalk:command",
      "OptionName": "IgnoreHealthCheck",
      "Value": "false"
    },
    {
      "Namespace": "aws:elasticbeanstalk:application",
      "OptionName": "Application Healthcheck URL",
      "Value": "TCP:80"
    }
  ],
  "PlatformArn": "arn:aws:elasticbeanstalk:us-east-1::platform/Tomcat 8 with Java 8 running on 64bit Amazon Linux/2.7.7",
}
```

```
"SolutionStackName": "64bit Amazon Linux 2017.09 v2.7.7 running Tomcat 8 Java 8",
>Status": "Ready",
Tier": {
  "Name": "WebServer"
  "Type": "Standard"
  "Version": "1.0"
},
"VersionLabel": "Sample Application"
}
```

The following is a list of valid value examples for `AwsElasticBeanstalkEnvironment` attributes:

- Status

Valid values: Aborting | Launching | Updating | LinkingFrom | LinkingTo | Ready | Terminating | Terminated

- Tier Name

Valid values: WebServer | Worker

- Tier Type

Valid values: Standard | SQS/HTTP

AwsElasticSearchDomain

The `AwsElasticSearchDomain` object provides details about an Amazon OpenSearch Service domain.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElasticSearchDomain` object. To view descriptions of `AwsElasticSearchDomain` attributes, see [AwsElasticSearchDomainDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsElasticSearchDomain": {
  "AccessPolicies": "string",
  "DomainStatus": {
    "DomainId": "string",
    "DomainName": "string",
    "Endpoint": "string",
    "Endpoints": {
      "string": "string"
    }
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": boolean,
    "TLSSecurityPolicy": "string"
  },
  "ElasticsearchClusterConfig": {
    "DedicatedMasterCount": number,
    "DedicatedMasterEnabled": boolean,
    "DedicatedMasterType": "string",
    "InstanceCount": number,
    "InstanceType": "string",
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": number
    },
    "ZoneAwarenessEnabled": boolean
  },
  "ElasticsearchVersion": "string",
  "EncryptionAtRestOptions": {
```

```

        "Enabled": boolean,
        "KmsKeyId": "string"
    },
    "LogPublishingOptions": {
        "AuditLogs": {
            "CloudWatchLogsLogGroupArn": "string",
            "Enabled": boolean
        },
        "IndexSlowLogs": {
            "CloudWatchLogsLogGroupArn": "string",
            "Enabled": boolean
        },
        "SearchSlowLogs": {
            "CloudWatchLogsLogGroupArn": "string",
            "Enabled": boolean
        }
    },
    "NodeToNodeEncryptionOptions": {
        "Enabled": boolean
    },
    "ServiceSoftwareOptions": {
        "AutomatedUpdateDate": "string",
        "Cancellable": boolean,
        "CurrentVersion": "string",
        "Description": "string",
        "NewVersion": "string",
        "UpdateAvailable": boolean,
        "UpdateStatus": "string"
    },
    "VPCOptions": {
        "AvailabilityZones": [
            "string"
        ],
        "SecurityGroupIds": [
            "string"
        ],
        "SubnetIds": [
            "string"
        ],
        "VPCId": "string"
    }
}

```

The following is a list of valid value examples for `AwsElasticSearchDomain` attributes:

- `ElasticsearchClusterConfig DedicatedMasterType`

The hardware configuration of the computer that hosts the dedicated master node. For example, `m3.medium.elasticsearch`. If this attribute is specified, then `DedicatedMasterEnabled` must be `true`.

Valid values: See [Supported instance types](#) in the *Amazon OpenSearch Service Developer Guide*.

- `ElasticsearchClusterConfig InstanceType`

Valid values: See [Supported instance types](#) in the *Amazon OpenSearch Service Developer Guide*.

- `ServiceSoftwareOptions UpdateStatus`

Valid values: `ELIGIBLE` | `PENDING_UPDATE` | `IN_PROGRESS` | `COMPLETED` | `NOT_ELIGIBLE`

AwsElbLoadBalancer

The `AwsElbLoadBalancer` object provides details about a Classic Load Balancer.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElbLoadBalancer` object. To view descriptions of `AwsElbLoadBalancer` attributes, see [AwsElbLoadBalancerDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsElbLoadBalancer": {
  "AvailabilityZones": ["us-west-2a"],
  "BackendServerDescriptions": [
    {
      "InstancePort": 80,
      "PolicyNames": ["doc-example-policy"]
    }
  ],
  "CanonicalHostedZoneName": "Z3DZXEOEXAMPLE",
  "CanonicalHostedZoneNameID": "my-load-balancer-444455556666.us-
west-2.elb.amazonaws.com",
  "CreatedTime": "2020-08-03T19:22:44.637Z",
  "DnsName": "my-load-balancer-444455556666.us-west-2.elb.amazonaws.com",
  "HealthCheck": {
    "HealthyThreshold": 2,
    "Interval": 30,
    "Target": "HTTP:80/png",
    "Timeout": 3,
    "UnhealthyThreshold": 2
  },
  "Instances": [
    {
      "InstanceId": "i-example"
    }
  ],
  "ListenerDescriptions": [
    {
      "Listener": {
        "InstancePort": 443,
        "InstanceProtocol": "HTTPS",
        "LoadBalancerPort": 443,
        "Protocol": "HTTPS",
        "SslCertificateId": "arn:aws:iam::444455556666:server-certificate/my-
server-cert"
      },
      "PolicyNames": ["ELBSecurityPolicy-TLS-1-2-2017-01"]
    }
  ],
  "LoadBalancerAttributes": {
    "AccessLog": {
      "EmitInterval": 60,
      "Enabled": true,
      "S3BucketName": "doc-example-bucket",
      "S3BucketPrefix": "doc-example-prefix"
    },
    "ConnectionDraining": {
      "Enabled": false,
      "Timeout": 300
    },
    "ConnectionSettings": {
      "IdleTimeout": 30
    },
    "CrossZoneLoadBalancing": {
      "Enabled": true
    },
    "AdditionalAttributes": [{
      "Key": "elb.http.desyncmitigationmode",
      "Value": "strictest"
    }]
  }
}
```

```

    },
    "LoadBalancerName": "example-load-balancer",
    "Policies": {
      "AppCookieStickinessPolicies": [
        {
          "CookieName": "",
          "PolicyName": ""
        }
      ],
      "LbCookieStickinessPolicies": [
        {
          "CookieExpirationPeriod": 60,
          "PolicyName": "my-example-cookie-policy"
        }
      ],
      "OtherPolicies": [
        "my-PublicKey-policy",
        "my-authentication-policy",
        "my-SSLNegotiation-policy",
        "my-ProxyProtocol-policy",
        "ELBSecurityPolicy-2015-03"
      ]
    },
    "Scheme": "internet-facing",
    "SecurityGroups": ["sg-example"],
    "SourceSecurityGroup": {
      "GroupName": "my-elb-example-group",
      "OwnerAlias": "444455556666"
    },
    "Subnets": ["subnet-example"],
    "VpcId": "vpc-a01106c2"
  }
}

```

AwsElbv2LoadBalancer

The `AwsElbv2LoadBalancer` object provides information about an Application Load Balancer, a Network Load Balancer, or a Gateway Load Balancer.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsElbv2LoadBalancer` object. To view descriptions of `AwsElbv2LoadBalancer` attributes, see [AwsElbv2LoadBalancerDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsElbv2LoadBalancer": {
  "AvailabilityZones": {
    "SubnetId": "string",
    "ZoneName": "string"
  },
  "CanonicalHostedZoneId": "string",
  "CreatedTime": "string",
  "DNSName": "string",
  "IpAddressType": "string",
  "LoadBalancerAttributes": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "Scheme": "string",
  "SecurityGroups": [ "string" ],
  "State": {

```

```
        "Code": "string",
        "Reason": "string"
    },
    "Type": "string",
    "VpcId": "string"
}
```

AwsIamAccessKey

The `AwsIamAccessKey` object contains details about an AWS Identity and Access Management access key that is related to a finding.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamAccessKey` object. To view descriptions of `AwsIamAccessKey` attributes, see [AwsIamAccessKeyDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsIamAccessKey": {
    "AccessKeyId": "string",
    "AccountId": "string",
    "CreatedAt": "string",
    "PrincipalId": "string",
    "PrincipalName": "string",
    "PrincipalType": "string",
    "SessionContext": {
        "Attributes": {
            "CreationDate": "string",
            "MfaAuthenticated": boolean
        },
        "SessionIssuer": {
            "AccountId": "string",
            "Arn": "string",
            "PrincipalId": "string",
            "Type": "string",
            "UserName": "string"
        }
    },
    "Status": "string"
}
```

AwsIamGroup

The `AwsIamGroup` object contains details about an IAM group.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamGroup` object. To view descriptions of `AwsIamGroup` attributes, see [AwsIamGroupDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsIamGroup": {
    "AttachedManagedPolicies": [
        {
            "PolicyArn": "arn:aws:iam::aws:policy/ExampleManagedAccess",
            "PolicyName": "ExampleManagedAccess",
        }
    ],
    "CreateDate": "2020-04-28T14:08:37.000Z",
    "GroupId": "AGPA4TPS3VLP7QEXAMPLE",
}
```

```
"GroupName": "Example_User_Group",
"GroupPolicyList": [
  {
    "PolicyName": "ExampleGroupPolicy"
  }
],
"Path": "/"
}
```

AwsIamPolicy

The `AwsIamPolicy` object represents an IAM permissions policy.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamPolicy` object. To view descriptions of `AwsIamPolicy` attributes, see [AwsIamPolicyDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsIamPolicy": {
  "AttachmentCount": 1,
  "CreateDate": "2017-09-14T08:17:29.000Z",
  "DefaultVersionId": "v1",
  "Description": "Example IAM policy",
  "IsAttachable": true,
  "Path": "/",
  "PermissionsBoundaryUsageCount": 5,
  "PolicyId": "ANPAJ2UCCR6DPCEXAMPLE",
  "PolicyName": "EXAMPLE-MANAGED-POLICY",
  "PolicyVersionList": [
    {
      "VersionId": "v1",
      "IsDefaultVersion": true,
      "CreateDate": "2017-09-14T08:17:29.000Z"
    }
  ],
  "UpdateDate": "2017-09-14T08:17:29.000Z"
}
```

AwsIamRole

The `AwsIamRole` object contains information about an IAM role, including all of the role's policies.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamRole` object. To view descriptions of `AwsIamRole` attributes, see [AwsIamRoleDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsIamRole": {
  "AssumeRolePolicyDocument": "{ 'Version': '2012-10-17', 'Statement': [ { 'Effect': 'Allow', 'Action': 'sts:AssumeRole' } ] }",
  "AttachedManagedPolicies": [
    {
      "PolicyArn": "arn:aws:iam::aws:policy/ExamplePolicy1",
      "PolicyName": "Example policy 1"
    },
    {
      "PolicyArn": "arn:aws:iam::444455556666:policy/ExamplePolicy2",
      "PolicyName": "Example policy 2"
    }
  ]
}
```

```

    }
  ],
  "CreateDate": "2020-03-14T07:19:14.000Z",
  "InstanceProfileList": [
    {
      "Arn": "arn:aws:iam::333333333333:ExampleProfile",
      "CreateDate": "2020-03-11T00:02:27Z",
      "InstanceProfileId": "AIPAIXEU4NUHUPEXAMPLE",
      "InstanceProfileName": "ExampleInstanceProfile",
      "Path": "/",
      "Roles": [
        {
          "Arn": "arn:aws:iam::444455556666:role/example-role",
          "AssumeRolePolicyDocument": "",
          "CreateDate": "2020-03-11T00:02:27Z",
          "Path": "/",
          "RoleId": "AROAJ52OTH4H7LEXAMPLE",
          "RoleName": "example-role",
        }
      ]
    }
  ],
  "MaxSessionDuration": 3600,
  "Path": "/",
  "PermissionsBoundary": {
    "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AdministratorAccess",
    "PermissionsBoundaryType": "PermissionsBoundaryPolicy"
  },
  "RoleId": "AROA4TPS3VLEXAMPLE",
  "RoleName": "BONESBootstrapHydra-OverbridgeOpsFunctionsLambda",
  "RolePolicyList": [
    {
      "PolicyName": "Example role policy"
    }
  ]
}

```

AwsIamUser

The `AwsIamUser` object provides information about an IAM user.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsIamUser` object. To view descriptions of `AwsIamUser` attributes, see [AwsIamUserDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsIamUser": {
  "AttachedManagedPolicies": [
    {
      "PolicyName": "ExamplePolicy",
      "PolicyArn": "arn:aws:iam::aws:policy/ExampleAccess"
    }
  ],
  "CreateDate": "2018-01-26T23:50:05.000Z",
  "GroupList": [],
  "Path": "/",
  "PermissionsBoundary": {
    "PermissionsBoundaryArn": "arn:aws:iam::aws:policy/AdministratorAccess",
    "PermissionsBoundaryType": "PermissionsBoundaryPolicy"
  },
  "UserId": "AIDACKCEVSQ6C2EXAMPLE",
  "UserName": "ExampleUser",
}

```

```
"UserPolicyList": [
  {
    "PolicyName": "InstancePolicy"
  }
]
```

AwsKinesisStream

The `AwsKinesisStream` object provides details about Amazon Kinesis Data Streams.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsKinesisStream` object. To view descriptions of `AwsKinesisStream` attributes, see [AwsKinesisStreamDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsKinesisStream": {
  "Name": "test-vir-kinesis-stream",
  "Arn": "arn:aws:kinesis:us-east-1:293279581038:stream/test-vir-kinesis-stream",
  "RetentionPeriodHours": 24,
  "ShardCount": 2,
  "StreamEncryption": {
    "EncryptionType": "KMS",
    "KeyId": "arn:aws:kms:us-east-1:293279581038:key/849cf029-4143-4c59-91f8-ea76007247eb"
  }
}
```

AwsKmsKey

The `AwsKmsKey` object provides details about an AWS KMS key.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsKmsKey` object. To view descriptions of `AwsKmsKey` attributes, see [AwsKmsKeyDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsKmsKey": {
  "AWSAccountId": "string",
  "CreationDate": "string",
  "Description": "string",
  "KeyId": "string",
  "KeyManager": "string",
  "KeyRotationStatus": boolean,
  "KeyState": "string",
  "Origin": "string"
}
```

The following is a list of valid value examples for `AwsKmsKey` attributes:

- `KeyState`

Valid values: `Enabled` | `Disabled` | `PendingDeletion` | `PendingImport` | `Unavailable`

AwsLambdaFunction

The `AwsLambdaFunction` object provides details about the configuration of an AWS Lambda function.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsLambdaFunction` object. To view descriptions of `AwsLambdaFunction` attributes, see [AwsLambdaFunctionDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsLambdaFunction": {
  "Code": {
    "S3Bucket": "string",
    "S3Key": "string",
    "S3ObjectVersion": "string",
    "ZipFile": "string"
  },
  "CodeSha256": "string",
  "DeadLetterConfig": {
    "TargetArn": "string"
  },
  "Environment": {
    "Variables": {
      "string": "string"
    }
  },
  "Error": {
    "ErrorCode": "string",
    "Message": "string"
  }
},
"FunctionName": "string",
"Handler": "string",
"KmsKeyArn": "string",
"LastModified": "string",
"Layers": {
  "Arn": "string",
  "CodeSize": number
},
"RevisionId": "string",
"Role": "string",
"Runtime": "string",
"Timeout": "integer",
"TracingConfig": {
  "Mode": "string"
},
"Version": "string",
"VpcConfig": {
  "SecurityGroupIds": [ "string" ],
  "SubnetIds": [ "string" ]
},
"MasterArn": "string",
"MemorySize": number
}
```

AwsLambdaLayerVersion

The `AwsLambdaLayerVersion` object provides details about a Lambda layer version.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsLambdaLayerVersion` object. To view descriptions of `AwsLambdaLayerVersion` attributes, see [AwsLambdaLayerVersionDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsLambdaLayerVersion": {
  "Version": 2,
  "CompatibleRuntimes": [
```

```
    "java8"  
  ],  
  "CreateDate": "2019-10-09T22:02:00.274+0000"  
}
```

AwsNetworkFirewallFirewall

The `AwsNetworkFirewallFirewall` object contains details about an AWS Network Firewall firewall.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsNetworkFirewallFirewall` object. To view descriptions of `AwsNetworkFirewallFirewall` attributes, see [AwsNetworkFirewallFirewallDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsNetworkFirewallFirewall": {  
  "DeleteProtection": false,  
  "FirewallArn": "arn:aws:network-firewall:us-east-1:024665936331:firewall/  
testfirewall",  
  "FirewallPolicyArn": "arn:aws:network-firewall:us-east-1:444455556666:firewall-policy/  
InitialFirewall",  
  "FirewallId": "dea7d8e9-ae38-4a8a-b022-672a830a99fa",  
  "FirewallName": "testfirewall",  
  "FirewallPolicyChangeProtection": false,  
  "SubnetChangeProtection": false,  
  "SubnetMappings": [  
    {  
      "SubnetId": "subnet-0183481095e588cdc"  
    },  
    {  
      "SubnetId": "subnet-01f518fad1b1c90b0"  
    }  
  ],  
  "VpcId": "vpc-40e83c38"  
}
```

AwsNetworkFirewallFirewallPolicy

The `AwsNetworkFirewallFirewallPolicy` object provides details about an AWS Network Firewall firewall policy. A firewall policy defines the behavior of a network firewall.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsNetworkFirewallFirewallPolicy` object. To view descriptions of `AwsNetworkFirewallFirewallPolicy` attributes, see [AwsNetworkFirewallFirewallPolicyDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsNetworkFirewallFirewallPolicy": {  
  "FirewallPolicy": {  
    "StatefulRuleGroupReferences": [  
      {  
        "ResourceArn": "arn:aws:network-firewall:us-east-1:444455556666:stateful-  
rulegroup/PatchesOnly"  
      }  
    ],  
    "StatelessDefaultActions": [ "aws:forward_to_sfe" ],  
    "StatelessFragmentDefaultActions": [ "aws:forward_to_sfe" ],  
    "StatelessRuleGroupReferences": [  
      {  
        "Priority": 1,  

```



```

        "ResourceArn": "arn:aws:network-firewall:us-east-1:444455556666:stateless-
rulegroup/Stateless-1"
    }
]
},
"FirewallPolicyArn": "arn:aws:network-firewall:us-east-1:444455556666:firewall-policy/
InitialFirewall",
"FirewallPolicyId": "9ceeda22-6050-4048-a0ca-50ce47f0cc65",
"FirewallPolicyName": "InitialFirewall",
"Description": "Initial firewall"
}

```

AwsNetworkFirewallRuleGroup

The `AwsNetworkFirewallRuleGroup` object provides details about an AWS Network Firewall rule group. Rule groups are used to inspect and control network traffic. *Stateless* rule groups apply to individual packets. *Stateful* rule groups apply to packets in the context of their traffic flow.

Rule groups are referenced in firewall policies.

The following examples show the AWS Security Finding Format (ASFF) for the `AwsNetworkFirewallRuleGroup` object. To view descriptions of `AwsNetworkFirewallRuleGroup` attributes, see [AwsNetworkFirewallRuleGroupDetails](#) in the *AWS Security Hub API Reference*.

Example – stateless rule group

```

"AwsNetworkFirewallRuleGroup": {
  "Capacity": 600,
  "RuleGroupArn": "arn:aws:network-firewall:us-east-1:444455556666:stateless-rulegroup/
Stateless-1",
  "RuleGroupId": "fb13c4df-b6da-4c1e-91ec-84b7a5487493",
  "RuleGroupName": "Stateless-1"
  "Description": "Example of a stateless rule group",
  "Type": "STATELESS",
  "RuleGroup": {
    "RulesSource": {
      "StatelessRulesAndCustomActions": {
        "CustomActions": [],
        "StatelessRules": [
          {
            "Priority": 1,
            "RuleDefinition": {
              "Actions": [
                "aws:pass"
              ],
              "MatchAttributes": {
                "DestinationPorts": [
                  {
                    "FromPort": 443,
                    "ToPort": 443
                  }
                ],
                "Destinations": [
                  {
                    "AddressDefinition": "192.0.2.0/24"
                  }
                ],
                "Protocols": [
                  6
                ],
                "SourcePorts": [
                  {
                    "FromPort": 0,
                    "ToPort": 65535
                  }
                ]
              }
            }
          }
        ]
      }
    }
  }
}

```

```

    ],
    "Sources": [
      {
        "AddressDefinition": "198.51.100.0/24"
      }
    ]
  }
}

```

Example – stateful rule group

```

"AwsNetworkFirewallRuleGroup": {
  "Capacity": 100,
  "RuleGroupArn": "arn:aws:network-firewall:us-east-1:444455556666:stateful-rulegroup/tupletest",
  "RuleGroupId": "38b71c12-da80-4643-a6c5-03337f8933e0",
  "RuleGroupName": "ExampleRuleGroup",
  "Description": "Example rule group",
  "Type": "STATEFUL",
  "RuleGroup": {
    "RuleSource": {
      "StatefulRules": [
        {
          "Action": "PASS",
          "Header": {
            "Destination": "Any",
            "DestinationPort": "443",
            "Direction": "ANY",
            "Protocol": "TCP",
            "Source": "Any",
            "SourcePort": "Any"
          },
          "RuleOptions": [
            {
              "Keyword": "sid:1"
            }
          ]
        }
      ]
    }
  }
}

```

The following is a list of valid value examples for `AwsNetworkFirewallRuleGroup` attributes:

- Action

Valid values: `PASS` | `DROP` | `ALERT`

- Protocol

Valid values: `IP` | `TCP` | `UDP` | `ICMP` | `HTTP` | `FTP` | `TLS` | `SMB` | `DNS` | `DCERPC` | `SSH` | `SMTP` | `IMAP` | `MSN` | `KRB5` | `IKEV2` | `TFTP` | `NTP` | `DHCP`

- Flags

Valid values: `FIN` | `SYN` | `RST` | `PSH` | `ACK` | `URG` | `ECE` | `CWR`

- Masks

Valid values: FIN | SYN | RST | PSH | ACK | URG | ECE | CWR

AwsOpenSearchServiceDomain

The `AwsOpenSearchServiceDomain` object contains information about an Amazon OpenSearch Service domain.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsOpenSearchServiceDomain` object. To view descriptions of `AwsOpenSearchServiceDomain` attributes, see [AwsOpenSearchServiceDomainDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsOpenSearchServiceDomain": {
  "AccessPolicies": "IAM_Id",
  "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserArn": "arn:aws:iam::123456789012:user/third-master-use",
      "MasterUserName": "third-master-use",
      "MasterUserPassword": "some-password"
    }
  },
  "Arn": "arn:aws:Opensearch:us-east-1:111122223333:somedomain",
  "ClusterConfig": {
    "InstanceType": "c5.large.search",
    "InstanceCount": 1,
    "DedicatedMasterEnabled": true,
    "ZoneAwarenessEnabled": false,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 2
    },
    "DedicatedMasterType": "c5.large.search",
    "DedicatedMasterCount": 3,
    "WarmEnabled": true,
    "WarmCount": 3,
    "WarmType": "ultrawarm1.large.search"
  },
  "DomainEndpoint": "https://es-2021-06-23t17-04-qowmgghud5vofgb5e4wmi.eu-central-1.es.amazonaws.com",
  "DomainEndpointOptions": {
    "EnforceHTTPS": false,
    "TLSecurityPolicy": "Policy-Min-TLS-1-0-2019-07",
    "CustomEndpointCertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/bda1bfff1-79c0-49d0-abe6-50a15a7477d4",
    "CustomEndpointEnabled": true,
    "CustomEndpoint": "example.com"
  },
  "DomainEndpoints": {
    "vpc": "vpc-endpoint-h2dsd34efgyghrtguk5gt6j2foh4.us-east-1.es.amazonaws.com"
  },
  "DomainName": "my-domain",
  "EncryptionAtRestOptions": {
    "Enabled": false,
    "KmsKeyId": "1a2a3a4-1a2a-3a4a-5a6a-1a2a3a4a5a6a"
  },
  "EngineVersion": "7.1",
  "Id": "123456789012",
  "LogPublishingOptions": {
    "IndexSlowLogs": {
```

```

        "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-group:/
aws/aes/domains/es-index-slow-logs",
        "Enabled": true
    },
    "SearchSlowLogs": {
        "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-group:/
aws/aes/domains/es-slow-logs",
        "Enabled": true
    },
    "AuditLogs": {
        "CloudWatchLogsLogGroupArn": "arn:aws:logs:us-east-1:111122223333:log-group:/
aws/aes/domains/es-slow-logs",
        "Enabled": true
    }
},
"NodeToNodeEncryptionOptions": {
    "Enabled": true
},
"ServiceSoftwareOptions": {
    "AutomatedUpdateDate": "2022-04-28T14:08:37.000Z",
    "Cancellable": false,
    "CurrentVersion": "R20210331",
    "Description": "There is no software update available for this domain.",
    "NewVersion": "OpenSearch_1.0",
    "UpdateAvailable": false,
    "UpdateStatus": "COMPLETED",
    "OptionalDeployment": false
},
"VpcOptions": {
    "SecurityGroupIds": [
        "sg-2a3a4a5a"
    ],
    "SubnetIds": [
        "subnet-1a2a3a4a"
    ]
}
}

```

The following is a list of valid value examples for `AwsOpenSearchServiceDomain` attributes:

- `InstanceType`

Valid values: See [Supported instance types](#) in the *Amazon OpenSearch Service Developer Guide*.

- `UpdateStatus`

Valid values: `ELIGIBLE` | `PENDING_UPDATE` | `IN_PROGRESS` | `COMPLETED` | `NOT_ELIGIBLE`

AwsRdsDbCluster

The `AwsRdsDbCluster` object provides details about an Amazon RDS database cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbCluster` object. To view descriptions of `AwsRdsDbCluster` attributes, see [AwsRdsDbClusterDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsRdsDbCluster": {
    "AllocatedStorage": 1,
    "AvailabilityZones": [
        "us-east-1c",
        "us-east-1e",

```

```

        "us-east-1a"
    ],
    "BackupRetentionPeriod": 1,
    "DatabaseName": "",
    "Status": "modifying",
    "Endpoint": "database-3.cluster-example.us-east-1.rds.amazonaws.com",
    "ReaderEndpoint": "database-3.cluster-ro-example.us-east-1.rds.amazonaws.com",
    "CustomEndpoints": [],
    "MultiAz": false,
    "Engine": "aurora-mysql",
    "EngineVersion": "5.7.mysql_aurora.2.03.4",
    "Port": 3306,
    "MasterUsername": "admin",
    "PreferredBackupWindow": "04:52-05:22",
    "PreferredMaintenanceWindow": "sun:09:32-sun:10:02",
    "ReadReplicaIdentifiers": [],
    "VpcSecurityGroups": [
        {
            "VpcSecurityGroupId": "sg-example-1",
            "Status": "active"
        }
    ],
    "HostedZoneId": "ZONE1",
    "StorageEncrypted": true,
    "KmsKeyId": "arn:aws:kms:us-east-1:777788889999:key/key1",
    "DbClusterResourceId": "cluster-example",
    "AssociatedRoles": [
        {
            "RoleArn": "arn:aws:iam::777788889999:role/aws-service-role/rds.amazonaws.com/AWSServiceRoleForRDS",
            "Status": "PENDING"
        }
    ],
    "ClusterCreateTime": "2020-06-22T17:40:12.322Z",
    "EnabledCloudwatchLogsExports": [
        "audit",
        "error",
        "general",
        "slowquery"
    ],
    "EngineMode": "provisioned",
    "DeletionProtection": false,
    "HttpEndpointEnabled": false,
    "ActivityStreamStatus": "stopped",
    "CopyTagsToSnapshot": true,
    "CrossAccountClone": false,
    "DomainMemberships": [],
    "DbClusterParameterGroup": "cluster-parameter-group",
    "DbSubnetGroup": "subnet-group",
    "DbClusterOptionGroupMemberships": [],
    "DbClusterIdentifier": "database-3",
    "DbClusterMembers": [
        {
            "IsClusterWriter": true,
            "PromotionTier": 1,
            "DbInstanceIdentifier": "database-3-instance-1",
            "DbClusterParameterGroupStatus": "in-sync"
        }
    ],
    "IamDatabaseAuthenticationEnabled": false
}

```

The following is a list of valid value examples for `AwsRdsDbCluster` attributes:

- `ActivityStreamStatus`

Valid values: stopped | starting | started | stopping

- Engine

Valid values: aurora | aurora-mysql | aurora-postgresql

- EngineMode

Valid values: provisioned | serverless | parallelquery | global | multimaster

- AssociatedRoles Status

Valid values: ACTIVE | PENDING | INVALID

AwsRdsDbClusterSnapshot

The `AwsRdsDbClusterSnapshot` object contains information about an Amazon RDS database cluster snapshot.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbClusterSnapshot` object. To view descriptions of `AwsRdsDbClusterSnapshot` attributes, see [AwsRdsDbClusterSnapshotDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbClusterSnapshot": {
  "AvailabilityZones": [
    "us-east-1a",
    "us-east-1d",
    "us-east-1e"
  ],
  "SnapshotCreateTime": "2020-06-22T17:40:12.322Z",
  "Engine": "aurora",
  "AllocatedStorage": 0,
  "Status": "available",
  "Port": 0,
  "VpcId": "vpc-faf7e380",
  "ClusterCreateTime": "2020-06-12T13:23:15.577Z",
  "MasterUsername": "admin",
  "EngineVersion": "5.6.10a",
  "LicenseModel": "aurora",
  "SnapshotType": "automated",
  "PercentProgress": 100,
  "StorageEncrypted": true,
  "KmsKeyId": "arn:aws:kms:us-east-1:777788889999:key/key1",
  "DbClusterIdentifier": "database-2",
  "DbClusterSnapshotIdentifier": "rds:database-2-2020-06-23-03-52",
  "IamDatabaseAuthenticationEnabled": false
}
```

AwsRdsDbInstance

The `AwsRdsDbInstance` object provides details about an Amazon RDS database instance.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbInstance` object. To view descriptions of `AwsRdsDbInstance` attributes, see [AwsRdsDbInstanceDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsDbInstance": {
```

```

"AllocatedStorage": 20,
"AssociatedRoles": [],
"AutoMinorVersionUpgrade": true,
"AvailabilityZone": "us-east-1d",
"BackupRetentionPeriod": 7,
"CaCertificateIdentifier": "certificate1",
"CharacterSetName": "",
"CopyTagsToSnapshot": true,
"DbClusterIdentifier": "",
"DbInstanceArn": "arn:aws:rds:us-east-1:111122223333:db:database-1",
"DbInstanceClass": "db.t2.micro",
"DbInstanceIdentifier": "database-1",
"DbInstancePort": 0,
"DbInstanceStatus": "available",
"DbiResourceId": "db-EXAMPLE123",
"DbName": "",
"DbParameterGroups": [
  {
    "DbParameterGroupName": "default.mysql5.7",
    "ParameterApplyStatus": "in-sync"
  }
],
"DbSecurityGroups": [],

"DbSubnetGroup": {
  "DbSubnetGroupName": "my-group-123abc",
  "DbSubnetGroupDescription": "My subnet group",
  "VpcId": "vpc-example1",
  "SubnetGroupStatus": "Complete",
  "Subnets": [
    {
      "SubnetIdentifier": "subnet-123abc",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1d"
      },
      "SubnetStatus": "Active"
    },
    {
      "SubnetIdentifier": "subnet-456def",
      "SubnetAvailabilityZone": {
        "Name": "us-east-1c"
      },
      "SubnetStatus": "Active"
    }
  ],
  "DbSubnetGroupArn": ""
},
"DeletionProtection": false,
"DomainMemberships": [],
"EnabledCloudWatchLogsExports": [],
"Endpoint": {
  "address": "database-1.example.us-east-1.rds.amazonaws.com",
  "port": 3306,
  "hostedZoneId": "ZONEID1"
},
"Engine": "mysql",
"EngineVersion": "5.7.22",
"EnhancedMonitoringResourceArn": "arn:aws:logs:us-east-1:111122223333:log-
group:Example:log-stream:db-EXAMPLE1",
"IamDatabaseAuthenticationEnabled": false,
"InstanceCreateTime": "2020-06-22T17:40:12.322Z",
"Iops": "",
"KmsKeyId": "",
"LatestRestorableTime": "2020-06-24T05:50:00.000Z",
"LicenseModel": "general-public-license",

```

```

"ListenerEndpoint": "",
"MasterUsername": "admin",
"MaxAllocatedStorage": 1000,
"MonitoringInterval": 60,
"MonitoringRoleArn": "arn:aws:iam::111122223333:role/rds-monitoring-role",
"MultiAz": false,
"OptionGroupMemberships": [
  {
    "OptionGroupName": "default:mysql-5-7",
    "Status": "in-sync"
  }
],
"PreferredBackupWindow": "03:57-04:27",
"PreferredMaintenanceWindow": "thu:10:13-thu:10:43",
"PendingModifiedValues": {
  "DbInstanceClass": "",
  "AllocatedStorage": "",
  "MasterUserPassword": "",
  "Port": "",
  "BackupRetentionPeriod": "",
  "MultiAZ": "",
  "EngineVersion": "",
  "LicenseModel": "",
  "Iops": "",
  "DbInstanceIdentifier": "",
  "StorageType": "",
  "CaCertificateIdentifier": "",
  "DbSubnetGroupName": "",
  "PendingCloudWatchLogsExports": "",
  "ProcessorFeatures": []
},
"PerformanceInsightsEnabled": false,
"PerformanceInsightsKmsKeyId": "",
"PerformanceInsightsRetentionPeriod": "",
"ProcessorFeatures": [],
"PromotionTier": "",
"PubliclyAccessible": false,
"ReadReplicaDBClusterIdentifiers": [],
"ReadReplicaDBInstanceIdentifiers": [],
"ReadReplicaSourceDBInstanceIdentifier": "",
"SecondaryAvailabilityZone": "",
"StatusInfos": [],
"StorageEncrypted": false,
"StorageType": "gp2",
"TdeCredentialArn": "",
"Timezone": "",
"VpcSecurityGroups": [
  {
    "VpcSecurityGroupId": "sg-example1",
    "Status": "active"
  }
]
}

```

AwsRdsDbSecurityGroup

The `AwsRdsDbSecurityGroup` object contains information about an Amazon Relational Database Service security group.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbSecurityGroup` object. To view descriptions of `AwsRdsDbSecurityGroup` attributes, see [AwsRdsDbSecurityGroupDetails](#) in the *AWS Security Hub API Reference*.

Example


```

"AwsRdsDbSecurityGroup": {
  "DbSecurityGroupArn": "arn:aws:rds:us-west-1:111122223333:secgrp:default",
  "DbSecurityGroupDescription": "default",
  "DbSecurityGroupName": "mysecgroup",
  "Ec2SecurityGroups": [
    {
      "Ec2SecurityGroupOwnerId": "myec2group",
      "Ec2SecurityGroupName": "default",
      "Ec2SecurityGroupOwnerId": "987654321021",
      "Status": "authorizing"
    }
  ],
  "IpRanges": [
    {
      "CidrIp": "0.0.0.0/0",
      "Status": "authorizing"
    }
  ],
  "OwnerId": "123456789012",
  "VpcId": "vpc-1234567f"
}

```

AwsRdsDbSnapshot

The `AwsRdsDbSnapshot` object contains details about an Amazon RDS database cluster snapshot.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsDbSnapshot` object. To view descriptions of `AwsRdsDbSnapshot` attributes, see [AwsRdsDbSnapshotDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsRdsDbSnapshot": {
  "DbSnapshotIdentifier": "rds:database-1-2020-06-22-17-41",
  "DbInstanceIdentifier": "database-1",
  "SnapshotCreateTime": "2020-06-22T17:41:29.967Z",
  "Engine": "mysql",
  "AllocatedStorage": 20,
  "Status": "available",
  "Port": 3306,
  "AvailabilityZone": "us-east-1d",
  "VpcId": "vpc-example1",
  "InstanceCreateTime": "2020-06-22T17:40:12.322Z",
  "MasterUsername": "admin",
  "EngineVersion": "5.7.22",
  "LicenseModel": "general-public-license",
  "SnapshotType": "automated",
  "Iops": null,
  "OptionGroupName": "default:mysql-5-7",
  "PercentProgress": 100,
  "SourceRegion": null,
  "SourceDbSnapshotIdentifier": "",
  "StorageType": "gp2",
  "TdeCredentialArn": "",
  "Encrypted": false,
  "KmsKeyId": "",
  "Timezone": "",
  "IamDatabaseAuthenticationEnabled": false,
  "ProcessorFeatures": [],
  "DbiResourceId": "db-resourceexample1"
}

```

The following is a list of valid value examples for `AwsRdsDbSnapshot` attributes:

- Engine

Valid values: aurora | aurora-mysql | aurora-postgresql | mariadb | mysql | oracle-ee | oracle-se2 | oracle-se1 | oracle-se | c | sqlserver-ee | sqlserver-se | sqlserver-ex | sqlserver-web

- ProcessorFeatures Name

Valid values: coreCount | threadsPerCore

- StorageType

Valid values: standard | gp2 | io1

AwsRdsEventSubscription

The `AwsRdsEventSubscription` contains details about an Amazon Relational Database Service event notification subscription. The subscription allows Amazon RDS to post events to an Amazon Simple Notification Service topic.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRdsEventSubscription` object. To view descriptions of `AwsRdsEventSubscription` attributes, see [AwsRdsEventSubscriptionDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsRdsEventSubscription": {
  "CustSubscriptionId": "myawsuser-secgrp",
  "CustomerAwsId": "111111111111",
  "Enabled": true,
  "EventCategoriesList": [
    "configuration change",
    "failure"
  ],
  "EventSubscriptionArn": "arn:aws:rds:us-east-1:111111111111:es:my-instance-events",
  "SnsTopicArn": "arn:aws:sns:us-east-1:111111111111:myawsuser-RDS",
  "SourceIdsList": [
    "si-sample",
    "mysqldb-rr"
  ],
  "SourceType": "db-security-group",
  "Status": "creating",
  "SubscriptionCreationTime": "2021-06-27T01:38:01.090Z"
}
```

AwsRedshiftCluster

The `AwsRedshiftCluster` object contains details about an Amazon Redshift cluster.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsRedshiftCluster` object. To view descriptions of `AwsRedshiftCluster` attributes, see [AwsRedshiftClusterDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsRedshiftCluster": {
  "AllowVersionUpgrade": true,
  "AutomatedSnapshotRetentionPeriod": 1,
  "AvailabilityZone": "us-west-2d",
  "ClusterAvailabilityStatus": "Unavailable",
  "ClusterCreateTime": "2020-08-03T19:22:44.637Z",
}
```

```

"ClusterIdentifier": "redshift-cluster-1",
"ClusterNodes": [
  {
    "NodeRole": "LEADER",
    "PrivateIPAddress": "192.0.2.108",
    "PublicIPAddress": "198.51.100.29"
  },
  {
    "NodeRole": "COMPUTE-0",
    "PrivateIPAddress": "192.0.2.22",
    "PublicIPAddress": "198.51.100.63"
  },
  {
    "NodeRole": "COMPUTE-1",
    "PrivateIPAddress": "192.0.2.224",
    "PublicIPAddress": "198.51.100.226"
  }
],
"ClusterParameterGroups": [
  {
    "ClusterParameterStatusList": [
      {
        "ParameterName": "max_concurrency_scaling_clusters",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "enable_user_activity_logging",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "auto_analyze",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "query_group",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "datestyle",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "extra_float_digits",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "search_path",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "statement_timeout",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      },
      {
        "ParameterName": "wlm_json_configuration",
        "ParameterApplyStatus": "in-sync",
        "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
      }
    ]
  }
]

```

```

        {
            "ParameterName": "require_ssl",
            "ParameterApplyStatus": "in-sync",
            "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
        },
        {
            "ParameterName": "use_fips_ssl",
            "ParameterApplyStatus": "in-sync",
            "ParameterApplyErrorDescription": "parameterApplyErrorDescription"
        }
    ],
    "ParameterApplyStatus": "in-sync",
    "ParameterGroupName": "temp"
}
],
"ClusterPublicKey": "Ja1rXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY Amazon-Redshift",
"ClusterRevisionNumber": 17498,
"ClusterSecurityGroups": [
    {
        "ClusterSecurityGroupName": "default",
        "Status": "active"
    }
],
"ClusterSnapshotCopyStatus": {
    "DestinationRegion": "us-west-2",
    "ManualSnapshotRetentionPeriod": -1,
    "RetentionPeriod": 1,
    "SnapshotCopyGrantName": "snapshotCopyGrantName"
},
"ClusterStatus": "available",
"ClusterSubnetGroupName": "default",
"ClusterVersion": "1.0",
"DBName": "dev",
"DeferredMaintenanceWindows": [
    {
        "DeferMaintenanceEndTime": "2020-10-07T20:34:01.000Z",
        "DeferMaintenanceIdentifier": "deferMaintenanceIdentifier",
        "DeferMaintenanceStartTime": "2020-09-07T20:34:01.000Z"
    }
],
"ElasticIpStatus": {
    "ElasticIp": "203.0.113.29",
    "Status": "active"
},
"ElasticResizeNumberOfNodeOptions": "4",
"Encrypted": false,
"Endpoint": {
    "Address": "redshift-cluster-1.example.us-west-2.redshift.amazonaws.com",
    "Port": 5439
},
"EnhancedVpcRouting": false,
"ExpectedNextSnapshotScheduleTime": "2020-10-13T20:34:01.000Z",
"ExpectedNextSnapshotScheduleTimeStatus": "OnTrack",
"HsmStatus": {
    "HsmClientCertificateIdentifier": "hsmClientCertificateIdentifier",
    "HsmConfigurationIdentifier": "hsmConfigurationIdentifier",
    "Status": "applying"
},
"IamRoles": [
    {
        "ApplyStatus": "in-sync",
        "IamRoleArn": "arn:aws:iam::111122223333:role/RedshiftCopyUnload"
    }
],
"KmsKeyId": "kmsKeyId",
"LoggingStatus": {

```

```

    "BucketName": "test-bucket",
    "LastFailureMessage": "test message",
    "LastFailureTime": "2020-08-09T13:00:00.000Z",
    "LastSuccessfulDeliveryTime": "2020-08-08T13:00:00.000Z",
    "LoggingEnabled": true,
    "S3KeyPrefix": "/"
  },
  "MaintenanceTrackName": "current",
  "ManualSnapshotRetentionPeriod": -1,
  "MasterUsername": "awsuser",
  "NextMaintenanceWindowStartTime": "2020-08-09T13:00:00.000Z",
  "NodeType": "dc2.large",
  "NumberOfNodes": 2,
  "PendingActions": [],
  "PendingModifiedValues": {
    "AutomatedSnapshotRetentionPeriod": 0,
    "ClusterIdentifier": "clusterIdentifier",
    "ClusterType": "clusterType",
    "ClusterVersion": "clusterVersion",
    "EncryptionType": "None",
    "EnhancedVpcRouting": false,
    "MaintenanceTrackName": "maintenanceTrackName",
    "MasterUserPassword": "masterUserPassword",
    "NodeType": "dc2.large",
    "NumberOfNodes": 1,
    "PubliclyAccessible": true
  },
  "PreferredMaintenanceWindow": "sun:13:00-sun:13:30",
  "PubliclyAccessible": true,
  "ResizeInfo": {
    "AllowCancelResize": true,
    "ResizeType": "ClassicResize"
  },
  "RestoreStatus": {
    "CurrentRestoreRateInMegaBytesPerSecond": 15,
    "ElapsedTimeInSeconds": 120,
    "EstimatedTimeToCompletionInSeconds": 100,
    "ProgressInMegaBytes": 10,
    "SnapshotSizeInMegaBytes": 1500,
    "Status": "restoring"
  },
  "SnapshotScheduleIdentifier": "snapshotScheduleIdentifier",
  "SnapshotScheduleState": "ACTIVE",
  "VpcId": "vpc-example",
  "VpcSecurityGroups": [
    {
      "Status": "active",
      "VpcSecurityGroupId": "sg-example"
    }
  ]
}

```

The following is a list of valid value examples for `AwsRedshiftCluster` attributes:

- `ParameterApplyStatus`

Valid values: `in-sync` | `pending-reboot` | `applying` | `invalid-parameter` | `apply-deferred` | `apply-error` | `unknown-error`

AwsS3AccountPublicAccessBlock

`AwsS3AccountPublicAccessBlock` provides information about the Amazon S3 Public Access Block configuration for accounts.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3AccountPublicAccessBlock` object. To view descriptions of `AwsS3AccountPublicAccessBlock` attributes, see [AwsS3AccountPublicAccessBlockDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsS3AccountPublicAccessBlock": {
  "BlockPublicAcls": true,
  "BlockPublicPolicy": true,
  "IgnorePublicAcls": false,
  "RestrictPublicBuckets": true
}
```

AwsS3Bucket

The `AwsS3Bucket` object provides details about an Amazon S3 bucket.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3Bucket` object. To view descriptions of `AwsS3Bucket` attributes, see [AwsS3BucketDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsS3Bucket": {
  "OwnerId": "AIDACKCEVSQ6C2EXAMPLE",
  "OwnerName": "s3bucketowner",
  "CreatedAt": "2007-11-30T01:46:56.000Z",
  "ServerSideEncryptionConfiguration": {
    "Rules": [
      {
        "ApplyServerSideEncryptionByDefault": {
          "SSEAlgorithm": "AES256",
          "KMSEMasterKeyID": "12345678-abcd-abcd-abcd-123456789012"
        }
      }
    ]
  },
  "BucketLifecycleConfiguration": {
    "Rules": [
      {
        "AbortIncompleteMultipartUpload": {
          "DaysAfterInitiation": 5
        },
        "ExpirationDate": "2021-11-10T00:00:00.000Z",
        "ExpirationInDays": 365,
        "ExpiredObjectDeleteMarker": false,
        "Filter": {
          "Predicate": {
            "Operands": [
              {
                "Prefix": "tmp/",
                "Type": "LifecyclePrefixPredicate"
              },
              {
                "Tag": {
                  "Key": "ArchiveAge",
                  "Value": "9m"
                },
                "Type": "LifecycleTagPredicate"
              }
            ],
            "Type": "LifecycleAndOperator"
          }
        }
      }
    ]
  }
}
```

```
    }
  },
  "ID": "Move rotated logs to Glacier",
  "NoncurrentVersionExpirationInDays": -1,
  "NoncurrentVersionTransitions": [
    {
      "Days": 2,
      "StorageClass": "GLACIER"
    }
  ],
  "Prefix": "rotated/",
  "Status": "Enabled",
  "Transitions": [
    {
      "Date": "2020-11-10T00:00:00.000Z",
      "Days": 100,
      "StorageClass": "GLACIER"
    }
  ]
}
]
},
"PublicAccessBlockConfiguration": {
  "BlockPublicAcls": true,
  "BlockPublicPolicy": true,
  "IgnorePublicAcls": true,
  "RestrictPublicBuckets": true,
}
}
```

The following is a list of valid value examples for `AwsS3Bucket` attributes:

- Filter Type

Valid values: `LifecycleAndOperator` | `LifecycleOrOperator`

- Filter Operands Type

Valid values: `LifecyclePrefixPredicate` | `LifecycleTagPredicate`

- StorageClass

Valid values: `GLACIER` | `STANDARD_IA` | `ONEZONE_IA` | `INTELLIGENT_TIERING` | `DEEP_ARCHIVE`

- BucketNotificationConfiguration Type

Valid values: `LambdaConfiguration` | `QueueConfiguration` | `TopicConfiguration`

- Filter Name

Valid values: `prefix` | `suffix`

- BucketVersioningConfiguration Status

Valid values: `Enabled` | `Suspended`

- Protocol

Valid values: `http` | `https`

- SSEAlgorithm

Valid values: `AES256` | `aws:kms`

AwsS3Object

The `AwsS3Object` object provides information about an Amazon S3 object.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsS3Object` object. To view descriptions of `AwsS3Object` attributes, see [AwsS3ObjectDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsS3Object": {
  "ContentType": "text/html",
  "ETag": "\"30a6ec7e1a9ad79c203d05a589c8b400\"",
  "LastModified": "2012-04-23T18:25:43.511Z",
  "ServerSideEncryption": "aws:kms",
  "SSEKMSKeyId": "arn:aws:kms:us-west-2:123456789012:key/4dff8393-e225-4793-a9a0-608ec069e5a7",
  "VersionId": "ws31OurgOOjH_HH1lIxPE35P.MELYaYh"
}
```

AwsSecretsManagerSecret

The `AwsSecretsManagerSecret` object provides details about a Secrets Manager secret.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSecretsManagerSecret` object. To view descriptions of `AwsSecretsManagerSecret` attributes, see [AwsSecretsManagerSecretDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsSecretsManagerSecret": {
  "RotationRules": {
    "AutomaticallyAfterDays": 30
  },
  "RotationOccurredWithinFrequency": true,
  "KmsKeyId": "kmsKeyId",
  "RotationEnabled": true,
  "RotationLambdaArn": "arn:aws:lambda:us-west-2:777788889999:function:MyTestRotationLambda",
  "Deleted": false,
  "Name": "MyTestDatabaseSecret",
  "Description": "My test database secret"
}
```

AwsSnsTopic

The `AwsSnsTopic` object contains details about an Amazon Simple Notification Service topic.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSnsTopic` object. To view descriptions of `AwsSnsTopic` attributes, see [AwsSnsTopicDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsSnsTopic": {
  "ApplicationSuccessFeedbackRoleArn":
    "arn:aws:iam::123456789012:role/ApplicationSuccessFeedbackRoleArn",
  "FirehoseFailureFeedbackRoleArn": "arn:aws:iam::123456789012:role/FirehoseFailureFeedbackRoleArn",
  "FirehoseSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/FirehoseSuccessFeedbackRoleArn",
  "HttpFailureFeedbackRoleArn": "arn:aws:iam::123456789012:role/HttpFailureFeedbackRoleArn",
}
```



```
        "HttpSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/  
HttpSuccessFeedbackRoleArn",  
        "KmsMasterKeyId": "alias/ExampleAlias",  
        "Owner": "123456789012",  
        "SqsFailureFeedbackRoleArn": "arn:aws:iam::123456789012:role/  
SqsFailureFeedbackRoleArn",  
        "SqsSuccessFeedbackRoleArn": "arn:aws:iam::123456789012:role/  
SqsSuccessFeedbackRoleArn",  
        "Subscription": {  
            "Endpoint": "http://sampleendpoint.com",  
            "Protocol": "http"  
        },  
        "TopicName": "SampleTopic"  
    }  
}
```

AwsSqsQueue

The `AwsSqsQueue` object contains information about an Amazon Simple Queue Service queue.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSqsQueue` object. To view descriptions of `AwsSqsQueue` attributes, see [AwsSqsQueueDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsSqsQueue": {  
    "DeadLetterTargetArn": "string",  
    "KmsDataKeyReusePeriodSeconds": number,  
    "KmsMasterKeyId": "string",  
    "QueueName": "string"  
}
```

AwsSsmPatchCompliance

The `AwsSsmPatchCompliance` object provides information about the state of a patch on an instance based on the patch baseline that was used to patch the instance.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsSsmPatchCompliance` object. To view descriptions of `AwsSsmPatchCompliance` attributes, see [AwsSsmPatchComplianceDetails](#) in the *AWS Security Hub API Reference*.

Example

```
AwsSsmPatchCompliance: {  
    "Patch": {  
        "ComplianceSummary": {  
            "ComplianceType": "Patch",  
            "CompliantCriticalCount": 0,  
            "CompliantHighCount": 0,  
            "CompliantInformationalCount": 0,  
            "CompliantLowCount": 0,  
            "CompliantMediumCount": 0,  
            "CompliantUnspecifiedCount": 461,  
            "ExecutionType": "Command",  
            "NonCompliantCriticalCount": 0,  
            "NonCompliantHighCount": 0,  
            "NonCompliantInformationalCount": 0,  
            "NonCompliantLowCount": 0,  
            "NonCompliantMediumCount": 0,  
            "NonCompliantUnspecifiedCount": 0,  
            "OverallSeverity": "UNSPECIFIED",  
            "PatchBaselineId": "pb-0c5b2769ef7cbe587",  
        }  
    }  
}
```

```
        "PatchGroup": "ExamplePatchGroup",  
        "Status": "COMPLIANT"  
    }  
  }  
}
```

The following is a list of valid value examples for `AwsSsmPatchCompliance` attributes:

- `OverallSeverity`

Valid values: `CRITICAL` | `HIGH` | `MEDIUM` | `LOW` | `INFORMATIONAL` | `UNSPECIFIED`

- `Status`

Valid values: `COMPLIANT` | `NON_COMPLIANT` | `UNSPECIFIED_DATA`

AwsWafRateBasedRule

The `AwsWafRateBasedRule` object contains details about a rate-based rule for global resources. A rate-based rule provides settings to indicate when to allow, block, or count a request. Rate-based rules include the number of requests that arrive over a specified period of time.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRateBasedRule` object. To view descriptions of `AwsWafRateBasedRule` attributes, see [AwsWafRateBasedRuleDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRateBasedRule":{  
  "MatchPredicates": [{  
    "DataId" : "391b7a7e-5f00-40d2-b114-3f27ceacbbb0",  
    "Negated" : "True",  
    "Type" : "IPMatch" ,  
  }],  
  "MetricName" : "MetricName",  
  "Name" : "Test",  
  "RateKey" : "IP",  
  "RateLimit" : 235000,  
  "RuleId" : "5dfb4085-f103-4ec6-b39a-d4a0dae5f47f"  
}
```

The following is a list of valid value examples for `AwsWafRateBasedRule` attributes:

- `MatchPredicates` Type

Valid values: `IPMatch` | `ByteMatch` | `SqlInjectionMatch` | `GeoMatch` | `SizeConstraint` | `XssMatch` | `RegexMatch`

AwsWafRegionalRateBasedRule

The `AwsWafRegionalRateBasedRule` object contains details about a rate-based rule for Regional resources. A rate-based rule provides settings to indicate when to allow, block, or count a request. Rate-based rules include the number of requests that arrive over a specified period of time.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRegionalRateBasedRule` object. To view descriptions of `AwsWafRegionalRateBasedRule` attributes, see [AwsWafRegionalRateBasedRuleDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalRateBasedRule":{
  "MatchPredicates" : [{
    "DataId" : "391b7a7e-5f00-40d2-b114-3f27ceacbbb0",
    "Negated" : "True",
    "Type" : "IPMatch" ,
  }],
  "MetricName" : "MetricName",
  "Name" : "Test",
  "RateKey" : "IP",
  "RateLimit" : 235000,
  "RuleId" : "5dfb4085-f103-4ec6-b39a-d4a0dae5f47f"
}
```

The following is a list of valid value examples for `AwsWafRegionalRateBasedRule` attributes:

- `MatchPredicates Type`

Valid values: `IPMatch` | `ByteMatch` | `SqlInjectionMatch` | `GeoMatch` | `SizeConstraint` | `XssMatch` | `RegexMatch`

AwsWafRegionalRule

The `AwsWafRegionalRule` object provides details about an AWS WAF Regional rule . This rule identifies the web requests that you want to allow, block, or count.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRegionalRule` object. To view descriptions of `AwsWafRegionalRule` attributes, see [AwsWafRegionalRuleDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalRule": {
  "MetricName": "SampleWAF_Rule__Metric_1",
  "Name": "bb-waf-regional-rule-not-empty-conditions-compliant",
  "RuleId": "8f651760-24fa-40a6-a9ed-4b60f1de95fe",
  "PredicateList": [{
    "DataId": "127d9346-e607-4e93-9286-c1296fb5445a",
    "Negated": false,
    "Type": "GeoMatch"
  }]
}
```

AwsWafRegionalRuleGroup

The `AwsWafRegionalRuleGroup` object provides details about an AWS WAF Regional rule group. A rule group is a collection of predefined rules that you add to a web access control list (web ACL).

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafRegionalRuleGroup` object. To view descriptions of `AwsWafRegionalRuleGroup` attributes, see [AwsWafRegionalRuleGroupDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRegionalRuleGroup": {
  "MetricName": "SampleWAF_Metric_1",
  "Name": "bb-WAFClassicRuleGroupWithRuleCompliant",
  "RuleGroupId": "2012ca6d-e66d-4d9b-b766-bfb03ad77cfb",
  "Rules": [{
    "Action": {
```

```

        "Type": "ALLOW"
    }
  }],
  "Priority": 1,
  "RuleId": "cdd225da-32cf-4773-8dc5-3bca3ed9c19c",
  "Type": "REGULAR"
}

```

AwsWafRegionalWebAcl

`AwsWafRegionalWebAcl` provides details about a Regional web access control list (web ACL). A web ACL contains the rules that identify the requests that you want to allow, block, or count.

The following is an example `AwsWafRegionalWebAcl` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see [AwsWafRegionalWebAclDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsWafRegionalWebAcl": {
  "DefaultAction": "ALLOW",
  "MetricName": "web-regional-webacl-metric-1",
  "Name": "WebACL_123",
  "RulesList": [
    {
      "Action": {
        "Type": "Block"
      },
      "Priority": 3,
      "RuleId": "24445857-852b-4d47-bd9c-61f05e4d223c",
      "Type": "REGULAR",
      "ExcludedRules": [
        {
          "ExclusionType": "Exclusion",
          "RuleId": "Rule_id_1"
        }
      ],
      "OverrideAction": {
        "Type": "OVERRIDE"
      }
    }
  ],
  "WebAclId": "443c76f4-2e72-4c89-a2ee-389d501c1f67"
}

```

AwsWafRule

`AwsWafRule` provides information about an AWS WAF rule. A rule identifies the web requests that you want to allow, block, or count.

The following is an example `AwsWafRule` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see [AwsWafRuleDetails](#) in the *AWS Security Hub API Reference*.

Example

```

"AwsWafRule": {
  "MetricName": "AwsWafRule_Metric_1",
  "Name": "AwsWafRule_Name_1",
  "PredicateList": [{

```

```
    "DataId": "cdd225da-32cf-4773-1dc2-3bca3ed9c19c",
    "Negated": false,
    "Type": "GeoMatch"
  }],
  "RuleId": "8f651760-24fa-40a6-a9ed-4b60f1de953e"
}
```

AwsWafRuleGroup

`AwsWafRuleGroup` provides information about an AWS WAF rule group. A rule group is a collection of predefined rules that you add to a web access control list (web ACL).

The following is an example `AwsWafRuleGroup` finding in the AWS Security Finding Format (ASFF). To view descriptions of `AwsApiGatewayV2Stage` attributes, see [AwsWafRuleGroupDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsWafRuleGroup": {
  "MetricName": "SampleWAF_Metric_1",
  "Name": "bb-WAFRuleGroupWithRuleCompliant",
  "RuleGroupId": "2012ca6d-e66d-4d9b-b766-bfb03ad77cfb",
  "Rules": [{
    "Action": {
      "Type": "ALLOW",
    },
    "Priority": 1,
    "RuleId": "cdd225da-32cf-4773-8dc5-3bca3ed9c19c",
    "Type": "REGULAR"
  }]
}
```

AwsWafWebAcl

The `AwsWafWebAcl` object provides details about an AWS WAF web ACL.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsWafWebAcl` object. To view descriptions of `AwsWafWebAcl` attributes, see [AwsWafWebAclDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsWafWebAcl": {
  "DefaultAction": "ALLOW",
  "Name": "MyWafAcl",
  "Rules": [
    {
      "Action": {
        "Type": "ALLOW"
      },
      "ExcludedRules": [
        {
          "RuleId": "5432a230-0113-5b83-bbb2-89375c5bfa98"
        }
      ],
      "OverrideAction": {
        "Type": "NONE"
      },
      "Priority": 1,
      "RuleId": "5432a230-0113-5b83-bbb2-89375c5bfa98",
      "Type": "REGULAR"
    }
  ]
}
```

```
    }  
  ],  
  "WebAclId": "waf-1234567890"  
}
```

AwsXrayEncryptionConfig

The `AwsXrayEncryptionConfig` object contains information about the encryption configuration for AWS X-Ray.

The following example shows the AWS Security Finding Format (ASFF) for the `AwsXrayEncryptionConfig` object. To view descriptions of `AwsXrayEncryptionConfig` attributes, see [AwsXrayEncryptionConfigDetails](#) in the *AWS Security Hub API Reference*.

Example

```
"AwsXRayEncryptionConfig":{  
  "KeyId": "arn:aws:kms:us-east-2:222222222222:key/example-key",  
  "Status": "UPDATING",  
  "Type": "KMS"  
}
```

The following is a list of valid value examples for `AwsXRayEncryptionConfig` attributes:

- `Status`

Valid values: `UPDATING` | `ACTIVE`

- `Type`

Valid values: `NONE` | `KMS`

Container

Container details that are related to a finding.

The following example shows the AWS Security Finding Format (ASFF) for the `Container` object. To view descriptions of `Container` attributes, see [ContainerDetails](#) in the *AWS Security Hub API Reference*.

Example:

```
"Container": {  
  "ContainerRuntime": "docker",  
  "ImageId": "image12",  
  "ImageName": "1111111/  
knotejs@sha256:372131c9fef111111111111115f4ed3ea5f9dce4dc3bd34ce21846588a3",  
  "LaunchedAt": "2018-09-29T01:25:54Z",  
  "Name": "knote",  
  "Privileged": true,  
  "VolumeMounts": [{  
    "Name": "vol-03909e9",  
    "MountPath": "/mnt/etc"  
  }]  
}
```

Other

The `Other` object allows you to provide custom fields and values. You use the `Other` object in the following cases.

- The resource type does not have a corresponding `Details` object. To provide details for the resource, you use the `Other` object.
- The `Details` object for the resource type does not include all of the attributes that you want to populate. In this case, use the `Details` object for the resource type to populate the available attributes. Use the `Other` object to populate the attributes that are not in the type-specific object.
- The resource type is not one of the provided types. In this case, you set `Resource.Type` to `Other`, and use the `Other` object to populate the details.

Type: Map of up to 50 key-value pairs

Each key-value pair must meet the following requirements.

- The key must contain fewer than 128 characters.
- The value must contain fewer than 1,024 characters.

Severity

Details about the severity of the finding.

For [BatchImportFindings](#) requests, finding providers should use the `Severity` object under [FindingProviderFields](#) (p. 189).

The finding severity does not consider the criticality of the involved assets or the underlying resource. Criticality is defined as the level of importance of the resources that are associated with the finding. For example, a resource that is associated with a mission critical application versus one that is associated with nonproduction testing. To capture information about resource criticality, use the `Criticality` field.

The finding must have either `Label` or `Normalized` populated. `Label` is the preferred attribute. Security Hub populates `Normalized`, but does not otherwise use it. If neither attribute is populated, then the finding is invalid.

To view descriptions of `Severity` attributes, see [Severity](#) in the *AWS Security Hub API Reference*.

Guidance for assigning severity (AWS services and partners)

For findings generated by AWS services and third-party partner products, the severity is based on the following.

- **Most severe** – Findings that are associated with actual data loss or denial of service
- **Second-most severe** – Findings that are associated with an active compromise but that do not indicate that data loss or other negative effects have occurred
- **Third-most severe** – Findings that are associated with issues that indicate potential for a future compromise

We recommend that you use the following guidance when translating findings' native severity scores to the value of `Severity.Label` for the ASFF.

- Informational findings. This category may include a finding for a `PASSED`, `WARNING`, or `NOT AVAILABLE` check or a sensitive data identification.

Suggested severity: `INFORMATIONAL`

- Findings that are associated with issues that could result in future compromises. For example, this category may include vulnerabilities, configuration weaknesses, and exposed passwords.

This generally aligns to the `Software and Configuration Checks` namespace under a finding's type.

Suggested severity: `LOW`

- Findings that are associated with issues that indicate an active compromise, but no indication that an adversary completed their objectives. For example, this category may include malware activity, hacking activity, and unusual behavior detection.

This generally aligns to the `Threat Detections and Unusual Behavior` namespaces under a finding's type.

Suggested severity: `MEDIUM`

- Findings that indicate that an adversary completed their objectives, such as active data loss or compromise or a denial of service.

This generally aligns to the `Effects` namespace under a finding's type.

Suggested severity: `HIGH` or `CRITICAL`

How Security Hub maps Label and Normalized values

If a `BatchImportFindings` request for a new finding only provides `Label` or only provides `Normalized`, then Security Hub automatically populates the value of the other field.

`Label` is the preferred attribute. Security Hub populates `Normalized`, but does not otherwise use it.

Security Hub does not make any automatic updates based on changes from `BatchUpdateFindings`.

If `BatchImportFindings` provides `Normalized` and does not provide `Label`, `Label` is set automatically as follows.

Normalized	Label
0	INFORMATIONAL
1–39	LOW
40–69	MEDIUM
70–89	HIGH
90–100	CRITICAL

If `BatchImportFindings` only provides `Label` and does not provide `Normalized`, `Normalized` is set automatically as follows.

Label	Normalized
INFORMATIONAL	0
LOW	1
MEDIUM	40
HIGH	70

Label	Normalized
CRITICAL	90

Threats

The `Threats` object provides details about the threat detected by a finding.

Example

```
"Threats": [{
  "FilePaths": [{
    "FileName": "b.txt",
    "FilePath": "/tmp/b.txt",
    "Hash": "sha256",
    "ResourceId": "arn:aws:ec2:us-west-2:123456789012:volume/vol-032f3bdd89aee112f",
  }],
  "ItemCount": 3,
  "Name": "Iot.linux.mirai.vwisi",
  "Severity": "HIGH"
}]
```

ThreatIntelIndicators

This is an example of threat intelligence details that are related to a finding.

This object is an array of up to five threat intelligence indicator objects.

To view descriptions of `ThreatIntelIndicators` attributes, see [ThreatIntelIndicator](#) in the *AWS Security Hub API Reference*.

Example

```
"ThreatIntelIndicators": [
  {
    "Type": "IPV4_ADDRESS",
    "Value": "8.8.8.8",
    "Category": "BACKDOOR",
    "LastObservedAt": "2018-09-27T23:37:31Z",
    "Source": "Threat Intel Weekly",
    "SourceUrl": "http://threatintelweekly.org/backdoors/8888"
  }
]
```

Vulnerabilities

The `Vulnerabilities` object provides a list of vulnerabilities that are associated with the findings.

To view descriptions of `Vulnerabilities` attributes, see [Vulnerability](#) in the *AWS Security Hub API Reference*.

Example

```
"Vulnerabilities" : [
  {
    "Cvss": [
      {
        "BaseScore": 4.7,
```

```
        "BaseVector": "AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N",
        "Version": "V3"
    },
    {
        "BaseScore": 4.7,
        "BaseVector": "AV:L/AC:M/Au:N/C:C/I:N/A:N",
        "Version": "V2"
    }
],
"Id": "CVE-2020-12345",
"ReferenceUrls": [
    "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-12418",
    "http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-17563"
],
"RelatedVulnerabilities": ["CVE-2020-12345"],
"Vendor": {
    "Name": "Alas",
    "Url": "https://alas.aws.amazon.com/ALAS-2020-1337.html",
    "VendorCreatedAt": "2020-01-16T00:01:43Z",
    "VendorSeverity": "Medium",
    "VendorUpdatedAt": "2020-01-16T00:01:43Z"
},
"VulnerablePackages": [
    {
        "Architecture": "x86_64",
        "Epoch": "1",
        "Name": "openssl",
        "Release": "16.amzn2.0.3",
        "Version": "1.0.2k"
    }
]
}
```

Workflow

The `Workflow` object provides information about the status of the investigation into a finding.

It is not intended for finding providers. It is only to be used by customers and by remediation, orchestration, and ticketing tools used by customers.

The workflow status can only be updated using `BatchUpdateFindings`. Customers can also update it from the console. See [the section called "Setting the workflow status for findings" \(p. 130\)](#).

To view descriptions of `Workflow` attributes, see [Workflow](#) in the *AWS Security Hub API Reference*.

Types taxonomy for ASFF

The following information describes the first three levels of the `Types` path. In the list, the top-level bullets are namespaces, the second-level bullets are categories, and the third-level bullets are classifiers. We recommend that finding providers use defined namespaces to help sort and group findings. The defined categories and classifiers may also be used, but are not required. Only the software and configuration checks namespace has defined classifiers.

- Namespaces
 - Categories
 - Classifiers

A finding provider may define a partial path for namespace/category/classifier. For example, the following finding types are all valid.

- TTPs
- TTPs/Defense Evasion
- TTPs/Defense Evasion/CloudTrailStopped

The tactics, techniques, and procedures categories in the following list align to the [MITRE ATT&CK Matrix™](#). Unusual behaviors reflect general unusual behavior, such as general statistical anomalies, and are not aligned with a specific TTP. However, you could classify a finding with both unusual behaviors and TTPs finding types.

- Software and Configuration Checks
 - Vulnerabilities
 - CVE
 - AWS Security Best Practices
 - Network Reachability
 - Runtime Behavior Analysis
 - Industry and Regulatory Standards
 - AWS Foundational Security Best Practices
 - CIS Host Hardening Benchmarks
 - CIS AWS Foundations Benchmark
 - PCI-DSS
 - Cloud Security Alliance Controls
 - ISO 90001 Controls
 - ISO 27001 Controls
 - ISO 27017 Controls
 - ISO 27018 Controls
 - SOC 1
 - SOC 2
 - HIPAA Controls (USA)
 - NIST 800-53 Controls (USA)
 - NIST CSF Controls (USA)
 - IRAP Controls (Australia)
 - K-ISMS Controls (Korea)
 - MTCS Controls (Singapore)
 - FISC Controls (Japan)
 - My Number Act Controls (Japan)
 - ENS Controls (Spain)
 - Cyber Essentials Plus Controls (UK)
 - G-Cloud Controls (UK)
 - C5 Controls (Germany)
 - IT-Grundschutz Controls (Germany)
 - GDPR Controls (Europe)
 - TISAX Controls (Europe)
 - Patch Management
- TTPs
 - Initial Access
 - Execution
 - Persistence

- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Collection
- Command and Control
- Effects
 - Data Exposure
 - Data Exfiltration
 - Data Destruction
 - Denial of Service
 - Resource Consumption
- Unusual Behaviors
 - Application
 - Network Flow
 - IP address
 - User
 - VM
 - Container
 - Serverless
 - Process
 - Database
 - Data
- Sensitive Data Identifications
 - PII
 - Passwords
 - Legal
 - Financial
 - Security
 - Business

Insights in AWS Security Hub

An AWS Security Hub insight is a collection of related findings. It identifies a security area that requires attention and intervention. For example, an insight might point out EC2 instances that are the subject of findings that detect poor security practices. An insight brings together findings from across finding providers.

Each insight is defined by a group by statement and optional filters. The group by statement indicates how to group the matching findings, and identifies the type of item that the insight applies to. For example, if an insight is grouped by resource identifier, then the insight produces a list of resource identifiers. The optional filters identify the matching findings for the insight. For example, you might want to only see findings from specific providers or findings that are associated with specific types of resources.

Security Hub offers several built-in managed insights. You cannot modify or delete managed insights.

To track security issues that are unique to your AWS environment and usage, you can create custom insights.

An insight only returns results if you have enabled integrations or standards that produce matching findings. For example, the managed insight **29. Top resources by counts of failed CIS checks** only returns results if you enable the CIS AWS Foundations standard.

Topics

- [Viewing and filtering the list of insights \(p. 277\)](#)
- [Viewing and taking action on insight results and findings \(p. 277\)](#)
- [Managed insights \(p. 279\)](#)
- [Managing custom insights \(p. 286\)](#)

Viewing and filtering the list of insights

The **Insights** page displays the list of available insights.

By default, the list displays both managed and custom insights. To filter the insight list based on insight type, choose the insight type from the dropdown menu that is next to the filter field.

- To display all of the available insights, choose **All insights**. This is the default option.
- To display only managed insights, choose **Security Hub managed insights**.
- To display only custom insights, choose **Custom insights**.

You also can filter the insight list based on text in the insight name.

In the filter field, type the text to use to filter the list. The filter is not case sensitive. The filter looks for insights that contain the text anywhere in the insight name.

Viewing and taking action on insight results and findings

For each insight, AWS Security Hub first determines the findings that match the filter criteria, and then uses the grouping attribute to group the matching findings.

From the **Insights** console page, you can view and take action on the results and findings.

If you enable cross-Region aggregation, then in the aggregation Region, the results for managed insights include findings from the aggregation Region and the linked Regions. For custom insight results, if the insight does not filter by Region, then the results include findings from the aggregation Region and linked Regions.

In other Regions, the insight results are only for that Region.

For information on how to configure cross-Region aggregation, see [Cross-Region aggregation \(p. 111\)](#).

Viewing and taking action on insight results (console)

The insight results consist of a grouped list of the results for the insight. For example, if the insight is grouped by resource identifiers, then the insight results are the list of resource identifiers. Each item in the results list indicates the number of matching findings for that item.

Note that if the findings are grouped by resource identifier or resource type, then the results include all of the resources in the matching findings. This includes resources that have a different type from the resource type specified in the filter criteria. For example, an insight identifies findings that are associated with S3 buckets. If a matching finding contains both an S3 bucket resource and an IAM access key resource, then the insight results list both of those resources.

The results list is sorted from most to fewest matching findings.

Security Hub can only display 100 results. If there are more than 100 grouping values, you only see the first 100.

In addition to the results list, the insight results display a set of charts summarizing the number of matching findings for the following attributes.

- **Severity label** – Number of findings for each severity label
- **AWS account ID** – Top five account IDs for the matching findings
- **Resource type** – Top five resource types for the matching findings
- **Resource ID** – Top five resource IDs for the matching findings
- **Product name** – Top five finding providers for the matching findings

If you have configured custom actions, then you can send selected results to a custom action. The action must be associated with a CloudWatch rule for the Security Hub Insight Results event type. See [Automated response and remediation \(p. 604\)](#).

If you have not configured custom actions, then the **Actions** menu is disabled.

To display and take action on the list of insight results

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. To display the list of insight results, choose the insight name.
4. Select the check box for each result to send to the custom action.
5. From the **Actions** menu, choose the custom action.

Viewing insight results (Security Hub API, AWS CLI)

To view insight results, you can use an API call or the AWS Command Line Interface.

To view insight results (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [GetInsightResults](#) operation. To identify the insight to return results for, you need the insight ARN. To obtain the insight ARNs for custom insights, use the [GetInsights](#) operation.
- **AWS CLI** – At the command line, run the `get-insight-results` command.

```
aws securityhub get-insight-results --insight-arn <insight ARN>
```

Example:

```
aws securityhub get-insight-results --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Viewing findings for an insight result (console)

From the insight results list, you can display the list of findings for each result.

To display and take action on insight findings

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. To display the list of insight results, choose the insight name.
4. To display the list of findings for an insight result, choose the item from the results list.

The findings list shows the active findings for the selected insight result that have a workflow status of **NEW** or **NOTIFIED**.

From the findings list, you can perform the following actions.

- [Change the filters and grouping for the list \(p. 126\)](#)
- [View details for individual findings \(p. 128\)](#)
- [Update the workflow status of findings \(p. 130\)](#)
- [Send findings to custom actions \(p. 131\)](#)

Managed insights

AWS Security Hub provides several managed insights.

You cannot edit or delete Security Hub managed insights. You can [view and take action on the insight results and findings \(p. 277\)](#). You can also [use a managed insight as the basis for a new custom insight \(p. 288\)](#).

As with all insights, a managed insight only returns results if you have enabled product integrations or security standards that can produce matching findings.

For insights that are grouped by resource identifier, the results include the identifiers of all of the resources in the matching findings. This includes resources that have a different type from the resource type in the filter criteria. For example, insight 2 identifies findings that are associated with S3 buckets. If a matching finding contains both an S3 bucket resource and an IAM access key resource, then the insight results include both resources.

In the current release, Security Hub offers the following managed insights:

1. AWS resources with the most findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/1`

Grouped by: Resource identifier

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

2. S3 buckets with public write or read permissions

ARN: `arn:aws:securityhub:::insight/securityhub/default/10`

Grouped by: Resource identifier

Finding filters:

- Type starts with `Effects/Data Exposure`
- Resource type is `AwsS3Bucket`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

3. AMIs that are generating the most findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/3`

Grouped by: EC2 instance image ID

Finding filters:

- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

4. EC2 instances involved in known Tactics, Techniques, and Procedures (TTPs)

ARN: `arn:aws:securityhub:::insight/securityhub/default/14`

Grouped by: Resource ID

Finding filters:

- Type starts with `TTPs`
- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

5. AWS principals with suspicious access key activity

ARN: `arn:aws:securityhub:::insight/securityhub/default/9`

Grouped by: IAM access key principal name

Finding filters:

- Resource type is `AwsIamAccessKey`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

6. AWS resources instances that don't meet security standards / best practices

ARN: `arn:aws:securityhub:::insight/securityhub/default/6`

Grouped by: Resource ID

Finding filters:

- Type is Software and Configuration Checks/Industry and Regulatory Standards/AWS Security Best Practices
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

7. AWS resources associated with potential data exfiltration

ARN: `arn:aws:securityhub:::insight/securityhub/default/7`

Grouped by: Resource ID

Finding filters:

- Type starts with Effects/Data Exfiltration/
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

8. AWS resources associated with unauthorized resource consumption

ARN: `arn:aws:securityhub:::insight/securityhub/default/8`

Grouped by: Resource ID

Finding filters:

- Type starts with Effects/Resource Consumption
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

9. S3 buckets that don't meet security standards / best practice

ARN: `arn:aws:securityhub:::insight/securityhub/default/11`

Grouped by: Resource ID

Finding filters:

- Resource type is AwsS3Bucket
- Type is Software and Configuration Checks/Industry and Regulatory Standards/AWS Security Best Practices
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

10. S3 buckets with sensitive data

ARN: `arn:aws:securityhub:::insight/securityhub/default/12`

Grouped by: Resource ID

Finding filters:

- Resource type is AwsS3Bucket
- Type starts with Sensitive Data Identifications/
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

11. Credentials that may have leaked

ARN: `arn:aws:securityhub:::insight/securityhub/default/13`

Grouped by: Resource ID

Finding filters:

- Type starts with Sensitive Data Identifications/Passwords/
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

12. EC2 instances that have missing security patches for important vulnerabilities

ARN: `arn:aws:securityhub:::insight/securityhub/default/16`

Grouped by: Resource ID

Finding filters:

- Type starts with Software and Configuration Checks/Vulnerabilities/CVE
- Resource type is AwsEc2Instance
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

13. EC2 instances with general unusual behavior

ARN: `arn:aws:securityhub:::insight/securityhub/default/17`

Grouped by: Resource ID

Finding filters:

- Type starts with Unusual Behaviors
- Resource type is AwsEc2Instance
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED.

14. EC2 instances that have ports accessible from the Internet

ARN: `arn:aws:securityhub:::insight/securityhub/default/18`

Grouped by: Resource ID

Finding filters:

- Type starts with Software and Configuration Checks/AWS Security Best Practices/Network Reachability
- Resource type is AwsEc2Instance
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

15. EC2 instances that don't meet security standards / best practices

ARN: `arn:aws:securityhub:::insight/securityhub/default/19`

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
 - Software and Configuration Checks/Industry and Regulatory Standards/
 - Software and Configuration Checks/AWS Security Best Practices
- Resource type is AwsEc2Instance

- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

16. EC2 instances that are open to the Internet

ARN: arn:aws:securityhub:::insight/securityhub/default/21

Grouped by: Resource ID

Finding filters:

- Type starts with Software and Configuration Checks/AWS Security Best Practices/Network Reachability
- Resource type is AwsEc2Instance
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

17. EC2 instances associated with adversary reconnaissance

ARN: arn:aws:securityhub:::insight/securityhub/default/22

Grouped by: Resource ID

Finding filters:

- Type starts with TTPs/Discovery/Recon
- Resource type is AwsEc2Instance
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

18. AWS resources that are associated with malware

ARN: arn:aws:securityhub:::insight/securityhub/default/23

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
 - Effects/Data Exfiltration/Trojan
 - TTPs/Initial Access/Trojan
 - TTPs/Command and Control/Backdoor
 - TTPs/Command and Control/Trojan
 - Software and Configuration Checks/Backdoor
 - Unusual Behaviors/VM/Backdoor
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

19. AWS resources associated with cryptocurrency issues

ARN: arn:aws:securityhub:::insight/securityhub/default/24

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
 - Effects/Resource Consumption/Cryptocurrency
 - TTPs/Command and Control/CryptoCurrency
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

20. AWS resources with unauthorized access attempts

ARN: arn:aws:securityhub:::insight/securityhub/default/25

Grouped by: Resource ID

Finding filters:

- Type starts with one of the following:
 - TTPs/Command and Control/UnauthorizedAccess
 - TTPs/Initial Access/UnauthorizedAccess
 - Effects/Data Exfiltration/UnauthorizedAccess
 - Unusual Behaviors/User/UnauthorizedAccess
 - Effects/Resource Consumption/UnauthorizedAccess
- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

21. Threat Intel indicators with the most hits in the last week

ARN: arn:aws:securityhub:::insight/securityhub/default/26

Finding filters:

- Created within the last 7 days

22. Top accounts by counts of findings

ARN: arn:aws:securityhub:::insight/securityhub/default/27

Grouped by: AWS account ID

Finding filters:

- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

23. Top products by counts of findings

ARN: arn:aws:securityhub:::insight/securityhub/default/28

Grouped by: Product name

Finding filters:

- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

24. Severity by counts of findings

ARN: arn:aws:securityhub:::insight/securityhub/default/29

Grouped by: Severity label

Finding filters:

- Record state is ACTIVE
- Workflow status is NEW or NOTIFIED

25. Top S3 buckets by counts of findings

ARN: arn:aws:securityhub:::insight/securityhub/default/30

Grouped by: Resource ID

Finding filters:

- Resource type is `AwsS3Bucket`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

26. Top EC2 instances by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/31`

Grouped by: Resource ID

Finding filters:

- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

27. Top AMIs by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/32`

Grouped by: EC2 instance image ID

Finding filters:

- Resource type is `AwsEc2Instance`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

28. Top IAM users by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/33`

Grouped by: IAM access key user name

Finding filters:

- Resource type is `AwsIamAccessKey`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

29. Top resources by counts of failed CIS checks

ARN: `arn:aws:securityhub:::insight/securityhub/default/34`

Grouped by: Resource ID

Finding filters:

- Generator ID starts with `arn:aws:securityhub:::ruleset/cis-aws-foundations-benchmark/v/1.2.0/rule`
- Updated in the last day
- Compliance status is `FAILED`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

30. Top integrations by counts of findings

ARN: `arn:aws:securityhub:::insight/securityhub/default/35`

Grouped by: Product ARN

Finding filters:

- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

31. Resources with the most failed security checks

ARN: `arn:aws:securityhub:::insight/securityhub/default/36`

Grouped by: Resource ID

Finding filters:

- Updated in the last day
- Compliance status is `FAILED`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

32. IAM users with suspicious activity

ARN: `arn:aws:securityhub:::insight/securityhub/default/37`

Grouped by: IAM user name

Finding filters:

- Resource type is `AwsIamUser`
- Record state is `ACTIVE`
- Workflow status is `NEW` or `NOTIFIED`

Managing custom insights

In addition to the AWS Security Hub managed insights, you can create custom insights to track issues and resources that are specific to your environment.

You can create completely new custom insights, or start from an existing custom or managed insight.

Each insight is configured with the following options.

- **Grouping attribute** – The grouping attribute determines which items are displayed in the insight results list. For example, if the grouping attribute is **Product name**, then the insight results display the number of findings that are associated with each finding provider.
- **Optional filters** – The filters narrow down the matching findings for the insight.

When querying your findings, Security Hub applies Boolean AND logic to the set of filters. In other words, a finding only matches if it matches all of the provided filters. For example, if the filters are "Product name is GuardDuty" and "Resource type is `AwsS3Bucket`," then matching findings must match both of these criteria.

However, Security Hub applies Boolean OR logic to filters that use the same attribute but different values. For example, if the filters are "Product name is GuardDuty" and "Product name is Amazon Inspector," then a finding matches if it was generated by either GuardDuty or Amazon Inspector.

Note that if you use the resource identifier or resource type as the grouping attribute, then the insight results include all of the resources that are in the matching findings. The list is not limited to resources that match a resource type filter. For example, an insight identifies findings that are associated with S3 buckets, and groups those findings by resource identifier. A matching finding contains both an S3 bucket resource and an IAM access key resource. The insight results include both resources.

Creating a custom insight (console)

From the console, you can create a completely new insight.

To create a custom insight

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. Choose **Create insight**.
4. To select the grouping attribute for the insight:
 - a. Choose the search box to display the filter options.
 - b. Choose **Group by**.
 - c. Select the attribute to use to group the findings that are associated with this insight.
 - d. Choose **Apply**.
5. (Optional) Choose any additional filters to use for this insight. For each filter, define the filter criteria, and then choose **Apply**.
6. Choose **Create insight**.
7. Enter an **Insight name**, then choose **Create insight**.

Creating a custom insight (Security Hub API, AWS CLI)

To create a custom insight, you can use an API call or the AWS Command Line Interface.

To create a custom insight (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `CreateInsight` operation. When you create a custom insight, you must provide the name, the filters, and the grouping attribute.
- **AWS CLI** – At the command line, run the `create-insight` command.

```
aws securityhub create-insight --name <insight name> --filters <filter values> --group-by-attribute <attribute name>
```

Example

```
aws securityhub create-insight --filters '{"ResourceType": [{"Comparison": "EQUALS", "Value": "AwsIamRole"}], "SeverityLabel": [{"Comparison": "EQUALS", "Value": "CRITICAL"}]}' --group-by-attribute "ResourceId" --name "Critical role findings"
```

Modifying a custom insight (console)

You can modify an existing custom insight to change the grouping value and filters. After you make the changes, you can save the updates to the original insight, or save the updated version as a new insight.

To modify an insight

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. Choose the custom insight to modify.
4. Edit the insight configuration as needed.
 - To change the attribute used to group findings in the insight:
 - a. To remove the existing grouping, choose the **X** next to the **Group by** setting.
 - b. Choose the search box.

- c. Select the attribute to use for grouping.
- d. Choose **Apply**.
- To remove a filter from the insight, choose the circled **X** next to the filter.
- To add a filter to the insight:
 - a. Choose the search box.
 - b. Select the attribute and value to use as a filter.
 - c. Choose **Apply**.
5. When you complete the updates, choose **Save insight**.
6. When prompted, do one of the following:
 - To update the existing insight to reflect your changes, choose **Update <Insight_Name>** and then choose **Save insight**.
 - To create a new insight with the updates, choose **Save new insight**. Enter an **Insight name**, and then choose **Save insight**.

Modifying a custom insight (Security Hub API, AWS CLI)

To modify a custom insight, you can use an API call or the AWS Command Line Interface.

To modify a custom insight (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [UpdateInsight](#) operation. To identify the custom insight, you provide the insight ARN. To obtain the insight ARNs for custom insights, use the [GetInsights](#) operation. You can then update the name, the filters, and the grouping value.
- **AWS CLI** – At the command line, run the `update-insight` command.

```
aws securityhub update-insight --insight-arn <insight ARN> [--name <new name>] [--filters <new filters>] [--group-by-attribute <new grouping attribute>]
```

Example

```
aws securityhub update-insight --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/alb2c3d4-5678-90ab-cdef-EXAMPLE11111" --filters '{"ResourceType": [{"Comparison": "EQUALS", "Value": "AwsIamRole"}], "SeverityLabel": [{"Comparison": "EQUALS", "Value": "HIGH"}]}' --name "High severity role findings"
```

Creating a new custom insight from a managed insight (console)

You cannot save changes to or delete a managed insight. You can use a managed insight as the basis for a new custom insight.

To create a new custom insight from a managed insight

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. Choose the managed insight to work from.

4. Edit the insight configuration as needed.
 - To change the attribute used to group findings in the insight:
 - a. To remove the existing grouping, choose the **X** next to the **Group by** setting.
 - b. Choose the search box.
 - c. Select the attribute to use for grouping.
 - d. Choose **Apply**.
 - To remove a filter from the insight, choose the circled **X** next to the filter.
 - To add a filter to the insight:
 - a. Choose the search box.
 - b. Select the attribute and value to use as a filter.
 - c. Choose **Apply**.
5. When your updates are complete, choose **Create insight**.
6. When prompted, enter an **Insight name**, and then choose **Create insight**.

Deleting a custom insight (console)

When you no longer want a custom insight, you can delete it. You cannot delete managed insights.

To delete a custom insight

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. Locate the custom insight to delete.
4. For that insight, choose the more options icon (the three dots in the top-right corner of the card).
5. Choose **Delete**.

Deleting a custom insight (Security Hub API, AWS CLI)

To delete a custom insight, you can use an API call or the AWS Command Line Interface.

To delete a custom insight (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DeleteInsight` operation. To identify the custom insight to delete, you provide the insight ARN. To obtain the insight ARNs for custom insights, use the `GetInsights` operation.
- **AWS CLI** – At the command line, run the `delete-insight` command.

```
aws securityhub delete-insight --insight-arn <insight ARN>
```

Example

```
aws securityhub delete-insight --insight-arn "arn:aws:securityhub:us-west-1:123456789012:insight/123456789012/custom/alb2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Product integrations in AWS Security Hub

AWS Security Hub can aggregate security finding data from several AWS services and from supported AWS Partner Network (APN) security solutions. This aggregation provides a comprehensive view of security and compliance across your AWS environment.

You can also send findings that are generated from your own custom security products.

Important

From the supported AWS and partner product integrations, Security Hub receives and consolidates only findings that are generated after you enable Security Hub in your AWS accounts.

The service does not retroactively receive and consolidate security findings that were generated before you enabled Security Hub.

For details on how Security Hub charges for ingested findings, see [Security Hub pricing](#).

Topics

- [Managing product integrations \(p. 290\)](#)
- [Available AWS service integrations \(p. 293\)](#)
- [Available third-party partner product integrations \(p. 304\)](#)
- [Using custom product integrations to send findings to AWS Security Hub \(p. 321\)](#)

Managing product integrations

The **Integrations** page in the AWS Management Console provides access to all of the available AWS and third-party product integrations. The AWS Security Hub API also provides operations to allow you to manage integrations.

Note

Some integrations are not available in all Regions. If an integration is not supported in the current Region, it is not listed on the **Integrations** page.

See also [the section called "Integrations that are supported in China \(Beijing\) and China \(Ningxia\)" \(p. 14\)](#) and [the section called "Integrations that are supported in AWS GovCloud \(US-East\) and AWS GovCloud \(US-West\)" \(p. 14\)](#).

Viewing and filtering the list of integrations (console)

From the **Integrations** page, you can view and filter the list of integrations.

To view the list of integrations

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the Security Hub navigation pane, choose **Integrations**.

On the **Integrations** page, the integrations with other AWS services are listed first, followed by the integrations with third-party products.

For each integration, the **Integrations** page provides the following information.

- The name of the company

- The name of the product
- A description of the integration
- The categories that the integration applies to
- How to enable the integration
- The current status of the integration

You can filter the list by entering text from the following fields.

- Company name
- Product name
- Integration description
- Categories

Viewing information about product integrations (Security Hub API, AWS CLI)

To view information about product integrations, you can use an API call or the AWS Command Line Interface. You can display information about all product integrations, or information about the product integrations that you have enabled.

To view information about all available product integrations (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DescribeProducts` operation. To identify a specific product integration to return, use the `ProductArn` parameter to provide the integration ARN.
- **AWS CLI** – At the command line, run the `describe-products` command. To identify a specific product integration to return, provide the integration ARN.

```
aws securityhub describe-products --product-arn "<integrationARN>"
```

Example

```
aws securityhub describe-products --product-arn "arn:aws:securityhub:us-east-1::product/3coresec/3coresec"
```

To view information about product integrations you have enabled (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `ListEnabledProductsForImport` operation.
- **AWS CLI** – At the command line, run the `list-enabled-products-for-import` command.

```
aws securityhub list-enabled-products-for-import
```

Enabling an integration

On the **Integrations** page, each integration provides the required steps to enable the integration.

For most of the integrations with other AWS services, the only required step is to enable the other service. The integration information includes a link to the service home page. When you enable the other

service, a resource-level permission that allows Security Hub to receive findings from the service is then automatically created and applied.

For third-party product integrations, you may need to purchase the integration from the AWS Marketplace, and then configure the integration. The integration information provides links to perform those tasks.

If more than one version of a product is available in AWS Marketplace, select the version to subscribe to and then choose **Continue to Subscribe**. For example, some products offer a standard version and an AWS GovCloud (US) version.

When you enable a product integration, a resource policy is automatically attached to that product subscription. This resource policy defines the permissions that Security Hub needs to receive findings from that product.

Disabling and enabling the flow of findings from an integration (console)

On the **Integrations** page, for integrations that send findings, the **Status** information indicates whether you are currently accepting findings.

To stop accepting findings, choose **Stop accepting findings**.

To resume accepting findings, choose **Accept findings**.

Disabling the flow of findings from an integration (Security Hub API, AWS CLI)

To disable the flow of findings from an integration, you can use an API call or the AWS Command Line Interface.

To disable the flow of findings from an integration (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisableImportFindingsForProduct` operation. To identify the integration to disable, you need the ARN of your subscription. To obtain the subscription ARNs for your enabled integrations, use the `ListEnabledProductsForImport` operation.
- **AWS CLI** – At the command line, run the `disable-import-findings-for-product` command.

```
aws securityhub disable-import-findings-for-product --product-subscription-arn <subscription ARN>
```

Example

```
aws securityhub disable-import-findings-for-product --product-subscription-arn "arn:aws:securityhub:us-west-1:123456789012:product-subscription/crowdstrike/crowdstrike-falcon"
```

Enabling the flow of findings from an integration (Security Hub API, AWS CLI)

To enable the flow of findings from an integration, you can use an API call or the AWS Command Line Interface.

To enable the flow of findings from an integration (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [EnableImportFindingsForProduct](#) operation. To enable Security Hub to receive findings from an integration, you need the product ARN. To obtain the ARNs for the available integrations, use the [DescribeProducts](#) operation.
- **AWS CLI**: At the command line, run the `enable-import-findings-for-product` command.

```
aws securityhub enable-import-findings-for-product --product-arn <integration ARN>
```

Example

```
aws securityhub enable-import-findings-for-product --product-arn "arn:aws:securityhub:us-east-1:123456789333:product/crowdstrike/crowdstrike-falcon"
```

Viewing the findings from an integration

For integrations that you accept findings for (**Status** is **Accepting findings**), to view a list of findings, choose **See findings**.

The findings list shows the active findings for the selected integration that have a workflow status of **NEW** or **NOTIFIED**.

If you enable cross-Region aggregation, then in the aggregation Region, the list includes findings from the aggregation Region and from linked Regions where the integration is enabled. Security Hub does not automatically enable integrations based on the cross-Region aggregation configuration.

In other Regions, the finding list for an integration only contains findings from the current Region.

For information on how to configure cross-Region aggregation, see [Cross-Region aggregation](#) (p. 111).

From the findings list, you can perform the following actions.

- [Change the filters and grouping for the list](#) (p. 126)
- [View details for individual findings](#) (p. 128)
- [Update the workflow status of findings](#) (p. 130)
- [Send findings to custom actions](#) (p. 131)

Available AWS service integrations

AWS Security Hub supports integrations with several AWS services.

Note

Some integrations are only available in select Regions.

If an integration is not supported, it is not listed on the **Integrations** page.

See also [Integrations that are supported in China \(Beijing\) and China \(Ningxia\)](#) (p. 14) and [Integrations that are supported in AWS GovCloud \(US-East\) and AWS GovCloud \(US-West\)](#) (p. 14).

With the exception of sensitive data findings from Amazon Macie, you're automatically opted in to all other AWS service integrations with Security Hub. If you've turned on Security Hub and the other service, no other step is needed to activate the integration between the two services.

The following sections provide details about each AWS service integration with Security Hub.

AWS Audit Manager (Receives findings)

AWS Audit Manager receives findings from Security Hub. These findings help Audit Manager users to prepare for audits.

To learn more about Audit Manager, see the [AWS Audit Manager User Guide](#). [AWS Security Hub checks supported by AWS Audit Manager](#) lists the controls for which Security Hub sends findings to Audit Manager.

AWS Chatbot (Sends findings)

AWS Chatbot is an interactive agent that helps you to monitor and interact with your AWS resources in your Slack channels and Amazon Chime chat rooms.

Security Hub sends findings to AWS Chatbot.

To learn more about the AWS Chatbot integration with Security Hub, see the [Security Hub integration overview](#) in the *AWS Chatbot Administrator Guide*.

AWS Config (Sends findings)

AWS Config is a service that allows you to assess, audit, and evaluate the configurations of your AWS resources. AWS Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations.

By using the integration with AWS Config, you can see the results of AWS Config managed and custom rule evaluations as findings in Security Hub. These findings can be viewed alongside other Security Hub findings, providing a comprehensive overview of your security posture.

AWS Config uses Amazon EventBridge to send AWS Config rule evaluations to Security Hub. Security Hub transforms the rule evaluations into findings that follow the [AWS Security Finding Format \(p. 132\)](#). Security Hub then enriches the findings on a best effort basis by getting more information about the impacted resources, such as the Amazon Resource Name (ARN) and creation date.

For more information about this integration, see the following sections.

How AWS Config sends findings to Security Hub

All findings in Security Hub use the standard JSON format of ASFF. ASFF includes details about the origin of the finding, the affected resource, and the current status of the finding. AWS Config sends managed and custom rule evaluations to Security Hub via EventBridge. Security Hub transforms the rule evaluations into findings that follow ASFF and enriches the findings on a best effort basis.

Types of findings that AWS Config sends to Security Hub

Once the integration is activated, AWS Config sends evaluations of all AWS Config managed rules and custom rules to Security Hub. Only evaluations from [service-linked AWS Config rules \(p. 324\)](#), such as those used to run checks on security controls, are excluded.

Sending AWS Config findings to Security Hub

When the integration is activated, Security Hub will automatically assign the permissions necessary to receive findings from AWS Config. Security Hub uses service-to-service level permissions that provide you with a safe way to activate this integration and import findings from AWS Config via Amazon EventBridge.

Latency for sending findings

When AWS Config creates a new finding, you can usually view the finding in Security Hub within five minutes.

Retrying when Security Hub is not available

AWS Config sends findings to Security Hub on a best-effort basis through EventBridge. When an event isn't successfully delivered to Security Hub, EventBridge retries delivery for up to 24 hours or 185 times, whichever comes first.

Updating existing AWS Config findings in Security Hub

After AWS Config sends a finding to Security Hub, it can send updates to the same finding to Security Hub to reflect additional observations of the finding activity.

Regions in which AWS Config findings exist

AWS Config findings occur on a Regional basis. AWS Config sends findings to Security Hub in the same Region or Regions where the findings occur.

Viewing AWS Config findings in Security Hub

To view your AWS Config findings, choose **Findings** from the Security Hub navigation pane. To filter the findings to display only AWS Config findings, choose **Product name** in the search bar drop down. Enter **Config**, and choose **Apply**.

Interpreting AWS Config finding names in Security Hub

Security Hub transforms AWS Config rule evaluations into findings that follow the [AWS Security Finding Format \(ASFF\)](#) (p. 132). AWS Config rule evaluations use a different event pattern compared to ASFF. The table below maps the AWS Config rule evaluation fields with their ASFF counterpart as they appear in Security Hub.

Config rule evaluation finding type	ASFF finding type	Hardcoded value
detail.awsAccountId	AwsAccountId	
detail.newEvaluationResult.resultRecord	CreatedTime	
detail.newEvaluationResult.resultRecord	UpdatedTime	
	ProductArn	"arn:<partition>;securityhub:<region>;product,aws/config"
	ProductName	"Config"
	CompanyName	"AWS"
	Region	"eu-central-1"
configRuleArn	GeneratorId, ProductFields	
detail.ConfigRuleARN/finding/hash	Id	
detail.configRuleName	Title, ProductFields	

Config rule evaluation finding type	ASFF finding type	Hardcoded value
detail.configRuleName	Description	"This finding is created for a resource compliance change for config rule: \${detail.ConfigRuleName}"
Configuration Item "ARN" or Security Hub computed ARN	Resources[i].id	
detail.resourceType	Resources[i].Type	"AwsS3Bucket"
	Resources[i].Partition	"aws"
	Resources[i].Region	"eu-central-1"
Configuration Item "configuration"	Resources[i].Details	
	SchemaVersion	"2018-10-08"
	Severity.Label	See "Interpreting Severity Label" below
	Types	["Software and Configuration Checks"]
detail.newEvaluationResult.compliance	Compliance.Status	"FAILED", "NOT_AVAILABLE", "PASSED", or "WARNING"

Interpreting severity label

All findings from AWS Config rule evaluations have a default severity label of **MEDIUM** in the ASFF. You can update the severity label of a finding with the [BatchUpdateFindings](#) API operation.

Typical finding from AWS Config

Security Hub transforms AWS Config rule evaluations into findings that follow the ASFF. The following is an example of a typical finding from AWS Config in the ASFF.

Note

If the description is more than 1024 characters, it will be truncated to 1024 characters and will say "(truncated)" at the end.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:config:eu-central-1:123456789012:config-rule/config-rule-mburzq/finding/45g070df80cb50b68fa6a43594kc6fda1e517932",
  "ProductArn": "arn:aws:securityhub:eu-central-1::product/aws/config",
  "ProductName": "Config",
  "CompanyName": "AWS",
  "Region": "eu-central-1",
  "GeneratorId": "arn:aws:config:eu-central-1:123456789012:config-rule/config-rule-mburzq",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks"
  ],
  "CreatedAt": "2022-04-15T05:00:37.181Z",
  "UpdatedAt": "2022-04-19T21:20:15.056Z",
```



```
"Severity": {
  "Label": "MEDIUM",
  "Normalized": 40
},
"Title": "s3-bucket-level-public-access-prohibited-config-integration-demo",
>Description": "This finding is created for a resource compliance change for config rule:
s3-bucket-level-public-access-prohibited-config-integration-demo",
"ProductFields": {
  "aws/securityhub/ProductName": "Config",
  "aws/securityhub/CompanyName": "AWS",
  "aws/securityhub/FindingId": "arn:aws:securityhub:eu-central-1::product/aws/
config/arn:aws:config:eu-central-1:123456789012:config-rule/config-rule-mburzq/
finding/46f070df80cd50b68fa6a43594dc5fda1e517902",
  "aws/config/ConfigRuleArn": "arn:aws:config:eu-central-1:123456789012:config-rule/config-
rule-mburzq",
  "aws/config/ConfigRuleName": "s3-bucket-level-public-access-prohibited-config-
integration-demo",
  "aws/config/ConfigComplianceType": "NON_COMPLIANT"
},
"Resources": [{
  "Type": "AwsS3Bucket",
  "Id": "arn:aws:s3:::config-integration-demo-bucket",
  "Partition": "aws",
  "Region": "eu-central-1",
  "Details": {
    "AwsS3Bucket": {
      "OwnerId": "4eddba300f1caa608fba2aad2c8fcfe30c32ca32777f64451eec4fb2a0f10d8c",
      "CreatedAt": "2022-04-15T04:32:53.000Z"
    }
  }
}],
"Compliance": {
  "Status": "FAILED"
},
"WorkflowState": "NEW",
"Workflow": {
  "Status": "NEW"
},
"RecordState": "ACTIVE",
"FindingProviderFields": {
  "Severity": {
    "Label": "MEDIUM"
  }
},
"Types": [
  "Software and Configuration Checks"
]
}
```

Enabling and configuring the integration

To use the AWS Config integration with Security Hub, you must set up both services and add at least one managed or custom rule in AWS Config. For information about how to set up AWS Config, see [Getting Started](#) in the *AWS Config Developer Guide*. For information about how to set up Security Hub, see [Setting up AWS Security Hub \(p. 66\)](#).

After you set up both AWS Config and Security Hub, the integration is activated automatically. AWS Config immediately begins to send findings to Security Hub.

Stopping the publication of findings to Security Hub

To stop sending findings to Security Hub, you can use the Security Hub console, the Security Hub API, or the AWS CLI.

See [Disabling and enabling the flow of findings from an integration \(console\)](#) (p. 292) or [Disabling the flow of findings from an integration \(Security Hub API, AWS CLI\)](#) (p. 292).

Amazon Detective (Linked from Security Hub)

Detective automatically collects log data from your AWS resources and uses machine learning, statistical analysis, and graph theory to help you visualize and conduct faster and more efficient security investigations.

The Security Hub integration with Detective allows you to pivot from Amazon GuardDuty findings in Security Hub into Detective. You can then use the Detective tools and visualizations to investigate them. The integration does not require any additional configuration in Security Hub or Detective.

For GuardDuty finding types, the finding details include an **Investigate in Detective** subsection. That subsection contains the link to Detective. See [Pivoting to an entity profile or finding overview from Amazon GuardDuty or AWS Security Hub](#) in the *Amazon Detective User Guide*.

If cross-Region aggregation is enabled, then when you pivot from the aggregation Region, Detective opens in the Region where the finding originated.

If a link does not work, then for troubleshooting advice, see [Troubleshooting the pivot](#).

AWS Firewall Manager (Sends findings)

Firewall Manager sends findings to Security Hub when a web application firewall (WAF) policy for resources or a web access control list (web ACL) rule is not in compliance. Firewall Manager also sends findings when AWS Shield Advanced is not protecting resources, or when an attack is identified.

If you are already using Firewall Manager, Security Hub automatically enables this integration. You do not need to take any additional action to begin to receive findings from Firewall Manager.

To learn more about the integration, view the **Integrations** page in the Security Hub console.

To learn more about Firewall Manager, see the [AWS WAF Developer Guide](#).

Amazon GuardDuty (Sends findings)

GuardDuty sends findings to Security Hub for all of the supported finding types.

New findings from GuardDuty are sent to Security Hub within five minutes. Updates to findings are sent based on the **Updated findings** setting for Amazon EventBridge in GuardDuty settings.

When you generate GuardDuty sample findings using the GuardDuty **Settings** page, Security Hub receives the sample findings and omits the prefix [Sample] in the finding type. For example, the sample finding type in GuardDuty [SAMPLE] Recon:IAMUser/ResourcePermissions is displayed as Recon:IAMUser/ResourcePermissions in Security Hub.

For more information about the GuardDuty integration, see [Integration with AWS Security Hub](#) in the *Amazon GuardDuty User Guide*.

AWS Health (Sends findings)

AWS Health provides ongoing visibility into your resource performance and the availability of your AWS services and accounts. You can use AWS Health events to learn how service and resource changes might affect your applications that run on AWS.

The integration with AWS Health does not use `BatchImportFindings`. Instead, AWS Health uses service-to-service event messaging to send findings to Security Hub.

For more information about the integration, see the following sections.

How AWS Health sends findings to Security Hub

In Security Hub, security issues are tracked as findings. Some findings come from issues that are detected by other AWS services or by third-party partners. Security Hub also has a set of rules that it uses to detect security issues and generate findings.

Security Hub provides tools to manage findings from across all of these sources. You can view and filter lists of findings and view details for a finding. See [Viewing finding lists and details in AWS Security Hub \(p. 126\)](#). You can also track the status of an investigation into a finding. See [Taking action on findings in AWS Security Hub \(p. 130\)](#).

All findings in Security Hub use a standard JSON format called the [AWS Security Finding Format \(ASFF\) \(p. 132\)](#). ASFF includes details about the source of the issue, the affected resources, and the current status of the finding.

AWS Health is one of the AWS services that sends findings to Security Hub.

Types of findings that AWS Health sends to Security Hub

Once the integration is enabled, AWS Health sends all security-related findings it generates to Security Hub. The findings are sent to Security Hub using the [AWS Security Finding Format \(ASFF\) \(p. 132\)](#). Security-related findings are defined as the following:

- Any finding associated with an AWS security service
- Any finding with the words `security`, `abuse`, or `certificate` in the AWS Health `typeCode`
- Any finding where the AWS Health service is `risk` or `abuse`

Sending AWS Health findings to Security Hub

When you choose to accept findings from AWS Health, Security Hub will automatically assign the permissions necessary to receive the findings from AWS Health. Security Hub uses service-to-service level permissions that provide you with a safe, easy way to enable this integration and import findings from AWS Health via Amazon EventBridge on your behalf. Choosing **Accept Findings** grants Security Hub permission to consume findings from AWS Health.

Latency for sending findings

When AWS Health creates a new finding, it is usually sent to Security Hub within five minutes.

Retrying when Security Hub is not available

AWS Health sends findings to Security Hub on a best-effort basis through EventBridge. When an event isn't successfully delivered to Security Hub, EventBridge retries sending the event for 24 hours.

Updating existing findings in Security Hub

After AWS Health sends a finding to Security Hub, it can send updates to the same finding to reflect additional observations of the finding activity to Security Hub.

Regions in which findings exist

For global events, AWS Health sends findings to Security Hub in `us-east-1` (AWS partition), `cn-northwest-1` (China partition), and `gov-us-west-1` (GovCloud partition). AWS Health sends Region-specific events to Security Hub in the same Region or Regions where the events occur.

Viewing AWS Health findings in Security Hub

To view your AWS Health findings in Security Hub, choose **Findings** from the navigation panel. To filter the findings to display only AWS Health findings, choose **Health** from the **Product name** field.

Interpreting AWS Health finding names in Security Hub

AWS Health sends the findings to Security Hub using the [AWS Security Finding Format \(ASFF\)](#) (p. 132). AWS Health finding uses a different event pattern compared to Security Hub ASFF format. The table below details all the AWS Health finding fields with their ASFF counterpart as they appear in Security Hub.

Health finding type	ASFF finding type	Hardcoded value
account	AwsAccountId	
detail.startTime	CreatedAt	
detail.eventDescription.latestDescription	Description	
detail.eventTypeCode	GeneratorId	
detail.eventArn (including account) + hash of detail.startTime	Id	
"arn:aws:securityhub:<region>::product/aws/health"	ProductArn	
account or resourceId	Resources[i].id	
	Resources[i].Type	"Other"
	SchemaVersion	"2018-10-08"
	Severity.Label	See "Interpreting Severity Label" below
"AWS Health -" detail.eventTypeCode	Title	
-	Types	["Software and Configuration Checks"]
event.time	UpdatedAt	
URL of the event on Health console	SourceUrl	

Interpreting severity label

The severity label in the ASFF finding is determined using the following logic:

- Severity **CRITICAL** if:
 - The `service` field in the AWS Health finding has the value `Risk`
 - The `typeCode` field in the AWS Health finding has the value `AWS_S3_OPEN_ACCESS_BUCKET_NOTIFICATION`
 - The `typeCode` field in the AWS Health finding has the value `AWS_SHIELD_INTERNET_TRAFFIC_LIMITATIONS_PLACED_IN_RESPONSE_TO_DDOS_ATTACK`
 - The `typeCode` field in the AWS Health finding has the value `AWS_SHIELD_IS_RESPONDING_TO_A_DDOS_ATTACK_AGAINST_YOUR_AWS_RESOURCES`

Severity **HIGH** if:

- The `service` field in the AWS Health finding has the value `Abuse`
- The `typeCode` field in the AWS Health finding contains the value `SECURITY_NOTIFICATION`
- The `typeCode` field in the AWS Health finding contains the value `ABUSE_DETECTION`

Severity **MEDIUM** if:

- The `service` field in the finding is any of the following: `ACM`, `ARTIFACT`, `AUDITMANAGER`, `BACKUP`, `CLOUDBACKUP`, `CLOUDHSM`, `CLOUDTRAIL`, `CLOUDWATCH`, `CODEGURU`, `COGNITO`, `CONFIG`, `CONTROLTOWER`, `DETECTIVE`, `DIRECTORYSERVICE`, `DRS`, `EVENTS`, `FIREWALLMANAGER`, `GUARDDUTY`, `IAM`, `INSPECTOR`, `INSPECTOR2`, `IOTDEVICEDEFENDER`, `KMS`, `MACIE`, `NETWORKFIREWALL`, `ORGANIZATIONS`, `RESILIENCEHUB`, `RESOURCEMANAGER`, `ROUTE53`, `SECURITYHUB`, `SECRETSMANAGER`, `SES`, `SHIELD`, `SSO`, or `WAF`
- The `typeCode` field in the AWS Health finding contains the value `CERTIFICATE`
- The `typeCode` field in the AWS Health finding contains the value `END_OF_SUPPORT`

Typical finding from AWS Health

AWS Health sends findings to Security Hub using the [AWS Security Finding Format \(ASFF\)](#) (p. 132). The following is an example of a typical finding from AWS Health.

Note

If the description is more than 1024 characters, it will be truncated to 1024 characters and will say (*truncated*) at the end.

```
{
  "SchemaVersion": "2018-10-08",
  "Id": "arn:aws:health:us-east-1:123456789012:event/SES/
AWS_SES_CMF_PENDING_TO_SUCCESS/
AWS_SES_CMF_PENDING_TO_SUCCESS_303388638044_33fe2115-8dad-40ce-
b533-78e29f49de96/101F7FBAEFC663977DA09CFF56A29236602834D2D361E6A8CA5140BFB3A69B30",
  "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/health",
  "GeneratorId": "AWS_SES_CMF_PENDING_TO_SUCCESS",
  "AwsAccountId": "123456789012",
  "Types": [
    "Software and Configuration Checks"
  ],
  "CreatedAt": "2022-01-07T16:34:04.000Z",
  "UpdatedAt": "2022-01-07T19:17:43.000Z",
  "Severity": {
    "Label": "MEDIUM",
    "Normalized": 40
  },
  "Title": "AWS Health - AWS_SES_CMF_PENDING_TO_SUCCESS",
  "Description": "Congratulations! Amazon SES has successfully detected the
MX record required to use 4557227d-9257-4e49-8d5b-18a99ced4be9.cmf.pinpoint.sysmon-
iad.adzel.com as a custom MAIL FROM domain for verified identity cmf.pinpoint.sysmon-
iad.adzel.com in AWS Region US East (N. Virginia).\n\nYou can now use this MAIL FROM
domain with cmf.pinpoint.sysmon-iad.adzel.com and any other verified identity that is
configured to use it. For information about how to configure a verified identity to use
a custom MAIL FROM domain, see http://docs.aws.amazon.com/SES/latest/DeveloperGuide/mail-
from-set.html .\n\nPlease note that this email only applies to AWS Region US East (N.
Virginia).",
  "SourceUrl": "https://phd.aws.amazon.com/phd/home#/event-log?
eventID=arn:aws:health:us-east-1::event/SES/AWS_SES_CMF_PENDING_TO_SUCCESS/
AWS_SES_CMF_PENDING_TO_SUCCESS_303388638044_33fe2115-8dad-40ce-b533-78e29f49de96",
  "ProductFields": {
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/
aws/health/arn:aws:health:us-east-1::event/SES/AWS_SES_CMF_PENDING_TO_SUCCESS/
AWS_SES_CMF_PENDING_TO_SUCCESS_303388638044_33fe2115-8dad-40ce-b533-78e29f49de96",
    "aws/securityhub/ProductName": "Health",
    "aws/securityhub/CompanyName": "AWS"
  },
}
```

```
    "Resources": [
      {
        "Type": "Other",
        "Id": "4557227d-9257-4e49-8d5b-18a99ced4be9.cmf.pinpoint.sysmon-
iad.adzel.com"
      }
    ],
    "WorkflowState": "NEW",
    "Workflow": {
      "Status": "NEW"
    },
    "RecordState": "ACTIVE",
    "FindingProviderFields": {
      "Severity": {
        "Label": "MEDIUM"
      },
      "Types": [
        "Software and Configuration Checks"
      ]
    }
  }
]
```

Enabling and configuring the integration

When you set up Security Hub, the integration with AWS Health is activated automatically. AWS Health immediately begins to send findings to Security Hub.

Stopping the publication of findings to Security Hub

To stop sending findings to Security Hub, you can use the Security Hub console, Security Hub API, or AWS CLI.

See [Disabling and enabling the flow of findings from an integration \(console\)](#) (p. 292) or [Disabling the flow of findings from an integration \(Security Hub API, AWS CLI\)](#) (p. 292).

IAM Access Analyzer (Sends findings)

With IAM Access Analyzer, all findings are sent to Security Hub.

IAM Access Analyzer uses logic-based reasoning to analyze resource-based policies that are applied to supported resources in your account. IAM Access Analyzer generates a finding when it detects a policy statement that allows an external principal access to a resource in your account.

To learn more, see [Integration with AWS Security Hub](#) in the *IAM User Guide*.

Amazon Inspector (Sends findings)

Amazon Inspector is a vulnerability management service that continuously scans your AWS workloads for vulnerabilities. Amazon Inspector automatically discovers and scans Amazon EC2 instances and container images that reside in the Amazon Elastic Container Registry. The scan looks for software vulnerabilities and unintended network exposure.

When you enable both Amazon Inspector and Security Hub, the integration is enabled automatically. Amazon Inspector begins to send findings to Security Hub. Amazon Inspector sends all of the findings it generates to Security Hub.

For more information about the integration, see [Integration with AWS Security Hub](#) in the *Amazon Inspector User Guide*.

Security Hub can also receive findings from Amazon Inspector Classic. Amazon Inspector Classic sends findings to Security Hub that are generated through assessment runs for all supported rules packages.

For more information about the integration, see [Integration with AWS Security Hub](#) in the *Amazon Inspector Classic User Guide*.

Findings for Amazon Inspector and Amazon Inspector Classic use the same product ARN. Amazon Inspector findings have the following entry in `ProductFields`:

```
"aws/inspector/ProductVersion": "2",
```

Amazon Macie (Sends findings)

A finding from Macie can indicate that there is a potential policy violation or that sensitive data, such as personally identifiable information (PII), is present in data that your organization stores in Amazon S3.

By default, Macie sends only policy findings to Security Hub. You can configure the integration to also send sensitive data findings to Security Hub.

In Security Hub, the finding type for a policy or sensitive data finding is changed to a value that is compatible with ASFF. For example, the `Policy:IAMUser/S3BucketPublic` finding type in Macie is displayed as `Effects/Data Exposure/Policy:IAMUser-S3BucketPublic` in Security Hub.

Macie also sends generated sample findings to Security Hub. For sample findings, the name of the affected resource is `macie-sample-finding-bucket` and the value for the `Sample` field is `true`.

For more information, see [Amazon Macie integration with AWS Security Hub](#) in the *Amazon Macie User Guide*.

AWS Systems Manager Explorer and OpsCenter (Receives and updates findings)

AWS Systems Manager Explorer and OpsCenter receive findings from Security Hub, and update those findings in Security Hub.

Explorer provides you with a customizable dashboard, providing key insights and analysis into the operational health and performance of your AWS environment.

OpsCenter provides you with a central location to view, investigate, and resolve operational work items.

For more information about Explorer and OpsCenter, see [Operations management](#) in the *AWS Systems Manager User Guide*.

AWS Systems Manager Patch Manager (Sends findings)

AWS Systems Manager Patch Manager sends findings to Security Hub when instances in a customer's fleet go out of compliance with their patch compliance standard.

Patch Manager automates the process of patching managed instances with both security related and other types of updates.

For more information about using Patch Manager, see [AWS Systems Manager Patch Manager](#) in the *AWS Systems Manager User Guide*.

AWS Trusted Advisor (Receives findings)

Trusted Advisor draws upon best practices learned from serving hundreds of thousands of AWS customers. Trusted Advisor inspects your AWS environment, and then makes recommendations when opportunities exist to save money, improve system availability and performance, or help close security gaps.

When you enable both Trusted Advisor and Security Hub, the integration is updated automatically.

Security Hub sends the results of its AWS Foundational Security Best Practices checks to Trusted Advisor.

For more information about the Security Hub integration with Trusted Advisor, see [Viewing AWS Security Hub controls in AWS Trusted Advisor](#) in the *AWS Support User Guide*.

Available third-party partner product integrations

AWS Security Hub is integrated with the following third-party products. For each provider, the list indicates how the integration interacts with findings. An integration can perform the following actions:

- Send findings that it generates to Security Hub.
- Receive findings from Security Hub.
- Update findings in Security Hub. Integrations that receive findings from Security Hub might also update those findings.

Integrations that send findings to Security Hub have an ARN.

Note

Some integrations are not available in all Regions.

If an integration is not supported, it is not listed on the console **Integrations** page

See also [the section called "Integrations that are supported in China \(Beijing\) and China \(Ningxia\)"](#) (p. 14) and [the section called "Integrations that are supported in AWS GovCloud \(US-East\) and AWS GovCloud \(US-West\)"](#) (p. 14).

If you have a security solution and are interested in becoming a Security Hub partner, send an email to [<securityhub-partners@amazon.com>](mailto:securityhub-partners@amazon.com). In the message, provide your company name, product name, AWS Partner Network (APN) tier level, and contact information.

To become a Security Hub partner, you must meet one of the following criteria:

- You are an AWS Select Tier Partner or above.
- You have joined the [AWS ISV Partner Path](#), and the product that you use for Security Hub integration has completed an [AWS Foundational Technical Review \(FTR\)](#). The product is then granted a "Reviewed by AWS" badge.

To get started, read through the [AWS Security Hub Partner Integration Guide](#). After you review the onboarding information, you can begin to work on your product manifest.

3CORESec – 3CORESec NTA (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/3coresec/3coresec`

3CORESec provides managed detection services for both on-premises and AWS systems. Their integration with Security Hub allows visibility into threats such as malware, privilege escalation, lateral movement, and improper network segmentation.

[Product link](#)

[Partner documentation](#)

Alert Logic – SIEMless Threat Management (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:733251395267:product/alertlogic/althreatmanagement`

Get the right level of coverage: vulnerability and asset visibility, threat detection and incident management, AWS WAF, and assigned SOC analyst options.

[Product link](#)

[Partner documentation](#)

Aqua Security – Aqua Cloud Native Security Platform (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/aquasecurity/aquasecurity`

Aqua Cloud Native Security Platform (CSP) provides full lifecycle security for container-based and serverless applications, from your CI/CD pipeline to runtime production environments.

[Product link](#)

[Partner documentation](#)

Aqua Security – Kube-bench (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/aqua-security/kube-bench`

Kube-bench is an open-source tool that runs the Center for Internet Security (CIS) Kubernetes Benchmark against your environment.

[Product link](#)

[Partner documentation](#)

Armor – Armor Anywhere (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:679703615338:product/armordefense/armoranywhere`

Armor Anywhere delivers managed security and compliance for AWS.

[Product link](#)

[Partner documentation](#)

Atlassian - Jira Service Management (Receives and updates findings)

Integration type: Receive and update

The AWS Service Management Connector for Jira sends findings from Security Hub to Jira. Jira issues are created based on the findings. When the Jira issues are updated, the corresponding findings are updated in Security Hub.

The integration only supports Jira Server and Jira Data Center.

For an overview of the integration and how it works, watch the video [AWS Security Hub – Bidirectional integration with Atlassian Jira Service Management](#).

[Product link](#)

[Partner documentation](#)

Atlassian – Opsgenie (Receives findings)

Integration type: Receive

Opsgenie is a modern incident management solution for operating always-on services, empowering development and operations teams to plan for service disruptions and stay in control during incidents.

Integrating with Security Hub ensures that mission critical security-related incidents are routed to the appropriate teams for immediate resolution.

[Product link](#)

[Partner documentation](#)

AttackIQ – AttackIQ (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/attackiq/attackiq-platform`

AttackIQ Platform emulates real adversarial behavior aligned with the MITRE ATT&CK Framework to help validate and improve your overall security posture.

[Product link](#)

[Partner documentation](#)

Barracuda Networks – Cloud Security Guardian (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:151784055945:product/barracuda/cloudsecurityguardian`

Barracuda Cloud Security Sentry helps organizations stay secure while building applications in, and moving workloads to, the public cloud.

[AWS Marketplace link](#)

[Product link](#)

BigID – BigID Enterprise (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/bigid/bigid-enterprise`

The BigID Enterprise Privacy Management Platform helps companies manage and protect sensitive data (PII) across all their systems.

[Product link](#)

[Partner documentation](#)

Blue Hexagon – Blue Hexagon forAWS (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/blue-hexagon/blue-hexagon-for-aws`

Blue Hexagon is a real time threat detection platform. It uses deep learning principles to detect known and unknown threats, including malware and network anomalies.

[Product link](#)

[Partner documentation](#)

Capitis Solutions – C2VS (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/capitis/c2vs`

C2VS is a customizable compliance solution designed to automatically identify your application-specific misconfigurations and their root cause.

[Product link](#)

[Partner documentation](#)

Caveonix – Caveonix Cloud (Sends and receives findings)

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/caveonix/caveonix-cloud`

Caveonix Cloud is a SaaS risk mitigation platform that delivers automated compliance and hybrid-cloud security posture management for comprehensive workload protection.

[Product link](#)

[Partner documentation](#)

Check Point – CloudGuard IaaS (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:758245563457:product/checkpoint/cloudguard-iaas`

Check Point CloudGuard easily extends comprehensive threat prevention security to AWS while protecting assets in the cloud.

[Product link](#)

[Partner documentation](#)

Check Point – CloudGuard Posture Management (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:634729597623:product/checkpoint/dome9-arc`

A SaaS platform that delivers verifiable cloud network security, advanced IAM protection, and comprehensive compliance and governance.

[Product link](#)

[Partner documentation](#)

Cloud Custodian – Cloud Custodian (Sends and receives findings)

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>::product/cloud-custodian/cloud-custodian`

Cloud Custodian enables users to be well managed in the cloud. The simple YAML DSL allows easily defined rules to enable a well-managed cloud infrastructure that's both secure and cost optimized.

[Product link](#)

[Partner documentation](#)

Cloud Storage Security – Antivirus for Amazon S3 (Sends findings)

Integration type: Send

Product ARN: arn:aws:securityhub:<REGION>::product/cloud-storage-security/antivirus-for-amazon-s3

Cloud Storage Security provides cloud native anti-malware and antivirus scanning for Amazon S3 objects.

Antivirus for Amazon S3 offers real time and scheduled scans of objects and files in Amazon S3 for malware and threats. It provides visibility and remediation for problem and infected files.

[Product link](#)

[Partner documentation](#)

cloudtamer.io – cloudtamer.io (Sends and receives findings)

Integration type: Send and receive

Product ARN: arn:aws:securityhub:<REGION>::product/cloudtamerio/cloudtamerio

cloudtamer.io is a complete cloud governance solution for AWS. cloudtamer.io gives stakeholders visibility into cloud operations and helps cloud users manage accounts, control budget and cost, and ensure continuous compliance.

[Product link](#)

[Partner documentation](#)

CrowdStrike – CrowdStrike Falcon (Sends findings)

Integration type: Send

Product ARN: arn:aws:securityhub:<REGION>:517716713836:product/crowdstrike/crowdstrike-falcon

The CrowdStrike Falcon single, lightweight sensor unifies next-generation antivirus, endpoint detection and response, and 24/7 managed hunting through the cloud.

[Product link](#)

[Partner documentation](#)

CyberArk – Privileged Threat Analytics (Sends findings)

Integration type: Send

Product ARN: arn:aws:securityhub:<REGION>:749430749651:product/cyberark/cyberark-pta

Privileged Threat Analytics collect, detect, alert, and respond to high-risk activity and behavior of privileged accounts to contain in-progress attacks.

[Product link](#)

[Partner documentation](#)

Data Theorem (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:product/data-theorem/api-cloud-web-secure`

Data Theorem continuously scans web applications, APIs, and cloud resources in search of security flaws and data privacy gaps to prevent AppSec data breaches.

[Product link](#)

[Partner documentation](#)

DisruptOps, Inc. – DisruptOPS (Sends and receives findings)

Integration type: Send and receive

Product ARN: `arn:aws:securityhub:<REGION>:product/disruptops-inc/disruptops`

The DisruptOps Security Operations Platform helps organizations maintain best security practices in your cloud through the use of automated guardrails.

[Product link](#)

[Partner documentation](#)

FireEye – FireEye Helix (Receives findings)

Integration type: Receive

FireEye Helix is a cloud-hosted security operations platform that allows organizations to take control of any incident from alert to fix.

[Product link](#)

[Partner documentation](#)

Forcepoint – Forcepoint CASB (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:365761988620:product/forcepoint/forcepoint-casb`

Forcepoint CASB allows you to discover cloud application use, analyze risk, and enforce appropriate controls for SaaS and custom applications.

[Product link](#)

[Partner documentation](#)

Forcepoint – Forcepoint Cloud Security Gateway (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:product/forcepoint/forcepoint-cloud-security-gateway`

Forcepoint Cloud Security Gateway is a converged cloud security service that provides visibility, control, and threat protection for users and data, wherever they are.

[Product link](#)

[Partner documentation](#)

Forcepoint – Forcepoint DLP (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:365761988620:product/forcepoint/forcepoint-dlp`

Forcepoint DLP addresses human-centric risk with visibility and control everywhere your people work and everywhere your data resides.

[Product link](#)

[Partner documentation](#)

Forcepoint – Forcepoint NGFW (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:365761988620:product/forcepoint/forcepoint-ngfw`

Forcepoint NGFW lets you connect your AWS environment into your enterprise network with the scalability, protection, and insights needed to manage your network and respond to threats.

[Product link](#)

[Partner documentation](#)

Fugue – Fugue (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:product/fugue/fugue`

Fugue is an agent-less, scalable cloud-native platform that automates the continuous validation of infrastructure-as-code and cloud runtime environments using the same policies.

[Product link](#)

[Partner documentation](#)

Guardicore – Centra 4.0 (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:324264561773:product/guardicore/guardicore`

Guardicore Centra provides flow visualization, micro-segmentation, and breach detection for workloads in modern data centers and clouds.

[Product link](#)

[Partner documentation](#)

Guardicore – Infection Monkey (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:324264561773:product/guardicore/aws-infection-monkey`

Infection Monkey is an attack simulation tool designed to test networks against attackers.

[Product link](#)

[Partner documentation](#)

HackerOne – Vulnerability Intelligence (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/hackerone/vulnerability-intelligence`

The HackerOne platform partners with the global hacker community to uncover the most relevant security issues. Vulnerability Intelligence enables your organization to go beyond automated scanning. It shares vulnerabilities that HackerOne ethical hackers have validated and provided steps to reproduce.

[Product link](#)

[Partner documentation](#)

Helecloud – Managed Security (Receives findings)

Integration type: Receive

HeleCloud is a Managed Services Provider, taking care of your AWS infrastructure so that you can focus on your core business.

[Product link](#)

IBM – QRadar (Receives findings)

Integration type: Receive

Product ARN: `arn:aws:securityhub:<REGION>:949680696695:product/ibm/qradar-siem`

IBM QRadar SIEM provides security teams with the ability to quickly and accurately detect, prioritize, investigate, and respond to threats.

[Product link](#)

[Partner documentation](#)

Juniper Networks – vSRX Next Generation Firewall (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/juniper-networks/vsrx-next-generation-firewall`

Juniper Networks' vSRX Virtual Next Generation Firewall delivers a complete cloud-based virtual firewall with advanced security, secure SD-WAN, robust networking, and built-in automation.

[AWS Marketplace link](#)

[Partner documentation](#)

[Product link](#)

k9 Security – Access Analyzer (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/k9-security/access-analyzer`

k9 Security notifies you when important access changes occur in your AWS Identity and Access Management account. With k9 Security, you can understand the access that each IAM user and role has to critical AWS services and your data.

k9 Security is built for continuous delivery, allowing you to operationalize IAM with actionable access audits and simple policy automation for AWS CDK and Terraform.

[Product link](#)

[Partner documentation](#)

Lacework (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/lacework/lacework`

Lacework is the data-driven security platform for the cloud. The Lacework Cloud Security Platform automates cloud security at scale so you can innovate with speed and safety.

[Product link](#)

[Partner documentation](#)

Logz.io Cloud SIEM (Receives findings)

Integration type: Receive

Logz.io is a provider of Cloud SIEM that provides advanced correlation of log and event data to help security teams to detect, analyze, and respond to security threats in real time.

[Product link](#)

[Partner documentation](#)

McAfee – MVISION Cloud Native Application Protection Platform (CNAPP) (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:297986523463:product/mcafee-skyhigh/mcafee-mvision-cloud-aws`

McAfee MVISION Cloud Native Application Protection Platform (CNAPP) offers Cloud Security Posture Management (CSPM) and Cloud Workload Protection Platform (CWPP) for your AWS environment.

[Product link](#)

[Partner documentation](#)

MicroFocus – MicroFocus Arcsight (Receives findings)

Integration type: Receive

ArcSight accelerates effective threat detection and response in real time, integrating event correlation and supervised and unsupervised analytics with response automation and orchestration.

[Product link](#)

[Partner documentation](#)

NETSCOUT – NETSCOUT Cyber Investigator (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:us-east-1::product/netscout/netscout-cyber-investigator`

NETSCOUT Cyber Investigator is an enterprise-wide network threat, risk investigation, and forensic analysis platform that helps to reduce the impact of cyber threats on businesses.

[Product link](#)

[Partner documentation](#)

PagerDuty – PagerDuty (Receives findings)

Integration type: Receive

The PagerDuty digital operations management platform empowers teams to proactively mitigate customer-impacting issues by automatically turning any signal into the right insight and action.

AWS users can use the PagerDuty set of AWS integrations to scale their AWS and hybrid environments with confidence.

When coupled with Security Hub aggregated and organized security alerts, PagerDuty allows teams to automate their threat response process and quickly set up custom actions to prevent potential issues.

PagerDuty users who are undertaking a cloud migration project can move quickly, while decreasing the impact of issues that occur throughout the migration lifecycle.

[Product link](#)

[Partner documentation](#)

Palo Alto Networks – Cortex XSOAR (Receives findings)

Integration type: Receive

Cortex XSOAR is a Security Orchestration, Automation, and Response (SOAR) platform that integrates with your entire security product stack to accelerate incident response and security operations.

[Product link](#)

[Partner documentation](#)

Palo Alto Networks – Prisma Cloud Compute (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:496947949261:product/twistlock/twistlock-enterprise`

Prisma Cloud Compute is a cloud native cybersecurity platform that protects VMs, containers, and serverless platforms.

[Product link](#)

[Partner documentation](#)

Palo Alto Networks – Prisma Cloud Enterprise (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:188619942792:product/paloaltonetworks/redlock`

Protects your AWS deployment with cloud security analytics, advanced threat detection, and compliance monitoring.

[Product link](#)

[Partner documentation](#)

Palo Alto Networks – VM-Series (Receives findings)

Integration type: Receive

Palo Alto VM-Series integration with Security Hub collects threat intelligence and sends it to the VM-Series next-generation firewall as an automatic security policy update that blocks malicious IP address activity.

[Product link](#)

[Partner documentation](#)

Prowler (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/prowler/prowler`

Prowler is an open source security tool to perform AWS checks related to security best practices, hardening, and continuous monitoring.

[Product link](#)

[Partner documentation](#)

Qualys – Vulnerability Management (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:805950163170:product/qualys/qualys-vm`

Qualys Vulnerability Management (VM) continuously scans and identifies vulnerabilities, protecting your assets.

[Product link](#)

[Partner documentation](#)

Rackspace Technology – Cloud Native Security (Receives findings)

Integration type: Receive

Rackspace Technology provides managed security services on top of native AWS security products for 24x7x365 monitoring by Rackspace SOC, advanced analysis, and threat remediation.

[Product link](#)

Rapid7 – InsightConnect (Receives findings)

Integration type: Receive

Rapid7 InsightConnect is a security orchestration and automation solution that enables your team to optimize SOC operations with little to no code.

[Product link](#)

[Partner documentation](#)

Rapid7 – InsightVM (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:336818582268:product/rapid7/insightvm`

Rapid7 InsightVM provides vulnerability management for modern environments, allowing you to efficiently find, prioritize, and remediate vulnerabilities.

[Product link](#)

[Partner documentation](#)

RSA – RSA Archer (Receives findings)

Integration type: Receive

RSA Archer IT and Security Risk Management allows you to determine which assets are critical to your business, establish and communicate security policies and standards, detect and respond to attacks, identify and remediate security deficiencies, and establish clear IT risk management best practices.

[Product link](#)

[Partner documentation](#)

SecureCloudDB – SecureCloudDB (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/secureclouddb/secureclouddb`

SecureCloudDB is a cloud native database security tool that provides comprehensive visibility of internal and external security postures and activity. It flags security violations and provides remediation on exploitable database vulnerabilities.

[Product link](#)

[Partner documentation](#)

SentinelOne (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/sentinelone/endpoint-protection`

SentinelOne is an autonomous extended detection and response (XDR) platform encompassing AI-powered prevention, detection, response, and hunting across endpoints, containers, cloud workloads, and IoT devices.

[AWS Marketplace link](#)

[Partner documentation](#)

[Product link](#)

ServiceNow – ITSM (Receives and updates findings)

Integration type: Receive and update

The ServiceNow integration with Security Hub allows security findings from Security Hub to be viewed within ServiceNow ITSM. You can also configure ServiceNow to automatically create an incident or problem when it receives a finding from Security Hub.

Any updates to these incidents and problems result in updates to the findings in Security Hub.

For an overview of the integration and how it works, watch the video [AWS Security Hub - Bidirectional integration with ServiceNow ITSM](#).

[Product link](#)

[Partner documentation](#)

Slack – Slack (Receives findings)

Integration type: Receive

Slack is a layer of the business technology stack that brings together people, data, and applications. It is a single place where people can effectively work together, find important information, and access hundreds of thousands of critical applications and services to do their best work.

[Product link](#)

[Partner documentation](#)

Sonrai Security – Sonrai Dig (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:product/sonrai-security/sonrai-dig`

Sonrai Dig monitors and remediates cloud misconfigurations and policy violations, so you can improve your security and compliance posture.

[Product link](#)

[Partner documentation](#)

Sophos – Server Protection (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:062897671886:product/sophos/sophos-server-protection`

Sophos Server Protection defends the critical applications and data at the core of your organization, using comprehensive defense-in-depth techniques.

[Product link](#)

[Partner documentation](#)

Splunk – Splunk Enterprise (Receives findings)

Integration type: Receive

Product ARN: `arn:aws:securityhub:<REGION>:112543817624:product/splunk/splunk-enterprise`

Splunk uses Amazon CloudWatch Events as a consumer of Security Hub findings. Send your data to Splunk for advanced security analytics and SIEM.

[Product link](#)

[Partner documentation](#)

Splunk – Splunk Phantom (Receives findings)

Integration type: Receive

With the Splunk Phantom application for AWS Security Hub, findings are sent to Phantom for automated context enrichment with additional threat intelligence information or to perform automated response actions.

[Product link](#)

[Partner documentation](#)

StackRox – StackRox Kubernetes Security (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/stackrox/kubernetes-security`

StackRox helps enterprises secure their container and Kubernetes deployments at scale by enforcing their compliance and security policies across the entire container life cycle – build, deploy, and run.

[Product link](#)

[Partner documentation](#)

Sumo Logic – Machine Data Analytics (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:956882708938:product/sumologicinc/sumologic-mda`

Sumo Logic is a secure, machine data analytics platform that enables development and security operations teams to build, run, and secure their AWS applications.

[Product link](#)

[Partner documentation](#)

Symantec – Cloud Workload Protection (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>:754237914691:product/symantec-corp/symantec-cwp`

Cloud Workload Protection provides complete protection for your Amazon EC2 instances with antimalware, intrusion prevention, and file integrity monitoring.

[Product link](#)

[Partner documentation](#)

Sysdig – Sysdig Secure for cloud (Sends findings)

Integration type: Send

Product ARN: arn:aws:securityhub:<REGION>::product/sysdig/sysdig-secure-for-cloud

Sysdig Secure for cloud supports asset discovery, risk management, Cloud Security Posture Management (CSPM), compliance, automatic vulnerability scanning for Amazon Elastic Container Registry (ECR) and Fargate, and threat detection based on CloudTrail. You can deploy all of these as a single security platform.

[Product link](#)

[Partner documentation](#)

Tenable – Tenable.io (Sends findings)

Integration type: Send

Product ARN: arn:aws:securityhub:<REGION>:422820575223:product/tenable/tenable-io

Accurately identify, investigate, and prioritize vulnerabilities. Managed in the cloud.

[Product link](#)

[Partner documentation](#)

ThreatModeler (Receives findings)

Integration type: Receive

ThreatModeler is an automated threat modeling solution that secures and scales the enterprise software and cloud development life cycle.

[Product link](#)

[Partner documentation](#)

Turbot – Turbot (Sends and receives findings)

Integration type: Send and receive

Product ARN: arn:aws:securityhub:<REGION>:453761072151:product/turbot/turbot

Turbot ensures that your cloud infrastructure is secure, compliant, scalable, and cost optimized.

[Product link](#)

[Partner documentation](#)

Vectra AI – Cognito Detect (Sends findings)

Integration type: Send

Product ARN: `arn:aws:securityhub:<REGION>::product/vectra-ai/cognito-detect`

Vectra is transforming cybersecurity by applying advanced AI to detect and respond to hidden cyberattackers before they can steal or cause damage.

[AWS Marketplace link](#)

[Partner documentation](#)

Using custom product integrations to send findings to AWS Security Hub

In addition to findings generated by the integrated AWS services and third-party products, Security Hub can also consume findings that are generated by other custom security products you may use.

You can send these findings into Security Hub manually using the `BatchImportFindings` API operation.

When setting up the custom integration, use the [guidelines and checklists](#) provided in the *Security Hub Partner Integration Guide*.

Requirements and recommendations for sending findings from custom security products

Before you can successfully invoke the `BatchImportFindings` API operation, you must enable Security Hub.

You must provide the finding details using the [the section called "Finding format" \(p. 132\)](#). For the findings from your custom integration, use the following requirements and recommendations.

Setting the product ARN

When you enable Security Hub, a default product Amazon Resource Name (ARN) for Security Hub is generated in your current account.

This product ARN has the following format: `arn:aws:securityhub:<region>:<account-id>:product/<account-id>/default`. For example, `arn:aws:securityhub:us-west-2:123456789012:product/123456789012/default`.

Use this product ARN as the value for the `ProductArn` attribute when invoking the `BatchImportFindings` API operation.

Defining the company and product name

You can use `BatchImportFindings` to set a preferred company name and product name for the custom integration that is sending findings to Security Hub.

Your specified names replace the preconfigured company name and product name, called personal name and default name respectively, and appear in the Security Hub console and the JSON of each finding. See [Using BatchImportFindings to create and update findings \(p. 119\)](#).

Setting the finding IDs

You must supply, manage, and increment your own finding IDs, using the `Id` attribute.

Each new finding must have a unique finding ID.

Setting the account ID

You must specify your own account ID, using the `AwsAccountId` attribute.

Setting the created at and updated at dates

You must supply your own timestamps for the `CreatedAt` and `UpdatedAt` attributes.

Updating findings from custom products

In addition to sending new findings from custom products, you can also use the `BatchImportFindings` API operation to update existing findings from custom products.

To update existing findings, use the existing finding ID (via the `Id` attribute). Resend the full finding with the appropriate information updated in the request, including a modified `UpdatedAt` timestamp.

Example custom integrations

You can use the following example custom product integrations as a guide to create your own custom solution.

Sending findings from Chef InSpec scans to Security Hub

You can create an AWS CloudFormation template that runs a `Chef InSpec` compliance scan and then sends findings to Security Hub.

For more details, see [Continuous compliance monitoring with Chef InSpec and AWS Security Hub](#).

Sending container vulnerabilities detected by Trivy to Security Hub

You can create an AWS CloudFormation template that uses `AquaSecurity Trivy` to scan containers for vulnerabilities, and then sends those vulnerability findings to Security Hub.

For more details, see [How to build a CI/CD pipeline for container vulnerability scanning with Trivy and AWS Security Hub](#).

Security standards and controls in AWS Security Hub

AWS Security Hub consumes, aggregates, and analyzes security findings from various supported AWS and third-party products.

Security Hub also generates its own findings by running automated and continuous checks against the rules in a set of supported security standards. These rules determine whether controls within a standard are being adhered to. The checks provide a readiness score and identify specific accounts and resources that require attention.

Security Hub provides controls for the following standards.

- [CIS AWS Foundations \(p. 346\)](#)
- [Payment Card Industry Data Security Standard \(PCI DSS\) \(p. 398\)](#)
- [AWS Foundational Security Best Practices \(p. 457\)](#)

For information on Security Hub pricing for security checks, see [Security Hub pricing](#).

Topics

- [How AWS Security Hub runs and uses security checks \(p. 323\)](#)
- [Viewing and managing security standards \(p. 333\)](#)
- [Viewing and managing controls \(p. 339\)](#)
- [Available security standards in AWS Security Hub \(p. 346\)](#)

How AWS Security Hub runs and uses security checks

For each enabled control, AWS Security Hub runs security checks. A security check determines whether your resources are in compliance with the control requirements.

Some checks run on a regular schedule. Other checks only run when there is a change to the resource state. See [the section called "Schedule for running security checks" \(p. 324\)](#).

Many security checks use AWS Config managed or custom rules to establish the compliance requirements. To run these checks, you must have AWS Config enabled. See [the section called "How Security Hub uses AWS Config rules to run security checks" \(p. 324\)](#). Others use custom Lambda functions, which are managed by Security Hub and are not visible to customers.

For each check, Security Hub creates or updates a finding. See [the section called "Generating and updating control findings" \(p. 325\)](#).

Security Hub uses the findings to assess your security posture for each control and across an entire standard. See [the section called "Determining the control status" \(p. 331\)](#) and [the section called "Determining the standard security score" \(p. 332\)](#).

Topics

- [How Security Hub uses AWS Config rules to run security checks \(p. 324\)](#)
- [Schedule for running security checks \(p. 324\)](#)
- [Generating and updating control findings \(p. 325\)](#)

- [Determining the overall status of a control from its findings \(p. 331\)](#)
- [Determining the security score for a security standard \(p. 332\)](#)

How Security Hub uses AWS Config rules to run security checks

To run security checks on your environment's resources, AWS Security Hub either uses steps specified by the standard, or uses specific AWS Config rules. Some rules are managed rules, which are managed by AWS Config. Other rules are custom rules that Security Hub develops.

AWS Config rules that Security Hub uses for controls are referred to as service-linked rules, because they are enabled and controlled by the Security Hub service.

To enable checks against these AWS Config rules, every account that has Security Hub enabled must first enable AWS Config, and enable resource recording for all resources. See [the section called "Enabling AWS Config" \(p. 8\)](#).

How Security Hub generates the service-linked rules

For every control that uses an AWS Config service-linked rule, Security Hub creates instances of the required rules in your AWS environment.

These service-linked rules are specific to Security Hub. It creates these service-linked rules even if other instances of the same rules already exist. The service-linked rule adds `securityhub` before the original rule name, and a unique identifier after the rule name. For example, for the original AWS Config managed rule `vpc-flow-logs-enabled`, the service-linked rule name would be something like `securityhub-vpc-flow-logs-enabled-12345`.

AWS Config has a [quota for the number of managed rules](#) per account per Region. The service-linked AWS Config rules that Security Hub creates do not count towards that quota. You can enable a security standard even if you have already reached the AWS Config limit for managed rules in your account. For service-linked rules, the quota is 250 rules per account per Region. This is in addition to the AWS Config quota on managed rules.

Viewing details about the AWS Config rules for controls

For controls that use AWS Config managed rules, the control description includes a link to the AWS Config rule details for the associated AWS Config rule. See [CIS AWS Foundations Benchmark controls \(p. 347\)](#), [the section called "PCI DSS controls" \(p. 399\)](#), and [the section called "Required AWS Config resources" \(p. 457\)](#). Custom rules are not linked from the control description.

For findings generated from those controls, the finding details include a link to the associated AWS Config rule. Note that to navigate to the AWS Config rule from finding details, you must also have an IAM permission in the selected account to navigate to AWS Config.

The finding details on the **Findings** page, **Insights** page, and **Integrations** page include a **Rules** link to the AWS Config rule details. See [the section called "Viewing finding details \(console\)" \(p. 128\)](#).

On the control details page, the **Investigate** column of the finding list contains a link to the AWS Config rule details. See [the section called "Viewing the AWS Config rule for a finding resource" \(p. 345\)](#).

Schedule for running security checks

After you enable a security standard, AWS Security Hub begins to run all checks within two hours. Most checks begin to run within 25 minutes. Until a control completes its first run of checks, its status is **No data**.

After the initial check, the schedule for each control can be either periodic or change triggered.

- Periodic checks run automatically within 12 hours after the most recent run. You cannot change the periodicity.
- Change-triggered checks run when the associated resource changes state. Even if the resource does not change state, the updated at time for change-triggered checks is refreshed every 18 hours. This helps to indicate that the control is still enabled.

In general, Security Hub uses change-triggered rules whenever possible. For a resource to use a change-triggered rule, it must support AWS Config configuration items.

For a control that is based on a managed AWS Config rule, the control description includes a link to the rule description in the *AWS Config Developer Guide*. That description includes whether the rule is change triggered or periodic.

Checks that use Security Hub custom Lambda functions are always periodic.

Generating and updating control findings

When it runs checks against the enabled controls for the enabled security standards, AWS Security Hub generates findings. These findings use the AWS Security Finding Format (ASFF). Note that if the finding size exceeds the maximum of 240 KB, then the `Resource.Details` object is removed. For findings for controls that use AWS Config rules, you can view the resource details on the AWS Config console.

Security Hub normally charges for each security check for a control. However, if multiple controls use the same AWS Config rule, then Security Hub only charges once for each check against the AWS Config rule. It generates separate findings for each control based on the check.

For example, the AWS Config rule `iam-password-policy` is used by multiple controls in the CIS AWS Foundations Benchmark standard and by IAM.7 in the Foundational Security Best Practices standard. Each time Security Hub runs a check against that AWS Config rule, it generates a separate finding for each related control, but only charges once for the check.

Compliance

For findings generated by security checks, the `Compliance` (p. 189) field in the ASFF contains the control-related findings details. The `Compliance` (p. 189) field includes the following information.

`RelatedRequirements`

The list of related requirements for the control. The requirements are from the third-party security framework for the control, such as the Payment Card Industry Data Security Standard.

`Status`

The result of the most recent check that Security Hub ran for a given control. The results of the previous checks are kept in an archived state for 90 days.

`StatusReasons`

Contains a list of reasons for the value of `Compliance.Status`. For each reason, `StatusReasons` includes the reason code and a description.

The following table lists the available status reason codes and descriptions. The remediation steps for a reason code depend on which control generated a finding with the reason code. For example, see remediation steps at [the section called "AWS Foundational Security Best Practices controls" \(p. 459\)](#).

Reason code	Compliance.Status	Description
CLOUDTRAIL_METRIC_FILTER_NOT_VALID	FAILED	The multi-Region CloudTrail trail does not have a valid metric filter.
CLOUDTRAIL_METRIC_FILTERS_NOT_PRESENT	FAILED	Metric filters are not present for the multi-Region CloudTrail trail.
CLOUDTRAIL_MULTI_REGION_NOT_PRESENT	FAILED	The account does not have a multi-Region CloudTrail trail with the required configuration.
CLOUDTRAIL_REGION_INVALID	WARNING	Multi-Region CloudTrail trails are not in the current Region.
CLOUDWATCH_ALARM_ACTIONS_NOT_VALID	FAILED	No valid alarm actions are present.
CLOUDWATCH_ALARMS_NOT_PRESENT	FAILED	CloudWatch alarms do not exist in the account.
CONFIG_ACCESS_DENIED	NOT_AVAILABLE AWS Config status is ConfigError	AWS Config access denied. Verify that AWS Config is enabled and has been granted sufficient permissions.
CONFIG_EVALUATIONS_EMPTY	PASSED	AWS Config evaluated your resources based on the rule. The rule did not apply to the AWS resources in its scope, the specified resources were deleted, or the evaluation results were deleted.
CONFIG_RETURNS_NOT_APPLICABLE	NOT_AVAILABLE	The compliance status is NOT_AVAILABLE because AWS Config returned a status of Not Applicable . AWS Config does not provide the reason for the status. Here are some possible reasons for the Not Applicable status: <ul style="list-style-type: none"> • The resource was removed from the scope of the AWS Config rule. • The AWS Config rule was deleted. • The resource was deleted. • The AWS Config rule logic can produce a Not Applicable status.
CONFIG_RULE_EVALUATION_ERROR	NOT_AVAILABLE AWS Config status is ConfigError	This reason code is used for several different types of evaluation errors. The description provides the specific reason information. The type of error can be one of the following:

Reason code	Compliance.Status	Description
		<ul style="list-style-type: none"> An inability to perform the evaluation because of a lack of permissions. The description provides the specific permission that is missing. A missing or invalid value for a parameter. The description provides the parameter and the requirements for the parameter value. An error reading from an S3 bucket. The description identifies the bucket and provides the specific error. A missing AWS subscription. A general timeout on the evaluation. A suspended account.
CONFIG_RULE_NOT_FOUND	NOT_AVAILABLE AWS Config status is ConfigError	The AWS Config rule is in the process of being created.
INTERNAL_SERVICE_ERROR	NOT_AVAILABLE	An unknown error occurred.
LAMBDA_CUSTOM_RUNTIME_DETAILS_NOT_AVAILABLE	NOT_AVAILABLE	Security Hub is unable to perform a check against a custom Lambda runtime.
S3_BUCKET_CROSS_ACCOUNT_CROSS_REGION	WARNING	<p>The finding is in a WARNING state, because the S3 bucket that is associated with this rule is in a different Region or account.</p> <p>This rule does not support cross-Region or cross-account checks.</p> <p>It is recommended that you disable this control in this Region or account. Only run it in the Region or account where the resource is located.</p>
SNS_SUBSCRIPTION_NOT_PRESENT	FAILED	The CloudWatch Logs metric filters do not have a valid Amazon SNS subscription.
SNS_TOPIC_CROSS_ACCOUNT	WARNING	<p>The finding is in a WARNING state.</p> <p>The SNS topic associated with this rule is owned by a different account. The current account cannot obtain the subscription information.</p> <p>The account that owns the SNS topic must grant to the current account the <code>sns:ListSubscriptionsByTopic</code> permission for the SNS topic.</p>

Reason code	Compliance.Status	Description
SNS_TOPIC_CROSS_ACCOUNT_CROSS_REGION	WARNING	<p>The finding is in a WARNING state because the SNS topic that is associated with this rule is in a different Region or account.</p> <p>This rule does not support cross-Region or cross-account checks.</p> <p>It is recommended that you disable this control in this Region or account. Only run it in the Region or account where the resource is located.</p>
SNS_TOPIC_INVALID	FAILED	The SNS topic associated with this rule is invalid.
THROTTLING_ERROR	NOT_AVAILABLE	The relevant API operation exceeded the allowed rate.

ProductFields

For findings generated by security checks, `ProductFields` includes the following fields:

`StandardsGuideArn` or `StandardsArn`

The ARN of the standard associated with the control.

For the CIS AWS Foundations Benchmark standard, the field is `StandardsGuideArn`.

For PCI DSS and AWS Foundational Security Best Practices standards, the field is `StandardsArn`.

`StandardsGuideSubscriptionArn` or `StandardsSubscriptionArn`

The ARN of the account's subscription to the standard.

For the CIS AWS Foundations Benchmark standard, the field is `StandardsGuideSubscriptionArn`.

For the PCI DSS and AWS Foundational Security Best Practices standards, the field is `StandardsSubscriptionArn`.

`RuleId` or `ControlId`

The identifier of the control.

For the CIS AWS Foundations Benchmark standard, the field is `RuleId`.

For the PCI DSS and AWS Foundational Security Best Practices standards, the field is `ControlId`.

`RecommendationUrl`

The URL to the remediation information for the control.

`RelatedAWSResources:0/name`

The name of the resource associated with the finding.

`RelatedAWSResource:0/type`

The type of resource associated with the control.

StandardsControlArn

The ARN of the control.

aws/securityhub/ProductName

For control-based findings, the product name is Security Hub.

aws/securityhub/CompanyName

For control-based findings, the company name is AWS.

aws/securityhub/annotation

A description of the issue uncovered by the control.

aws/securityhub/FindingId

The identifier of the finding.

Severity

The severity assigned to a Security Hub control identifies the importance of the control. It determines the severity label assigned to the control findings.

Severity criteria

The severity of a control is determined based on an assessment of the following criteria:

- **How difficult is it for a threat actor to take advantage of the configuration weakness associated with the control?**

The difficulty is determined by the amount of sophistication or complexity that is required to use the weakness to carry out a threat scenario.

- **How likely is it that the weakness will lead to a compromise of your AWS accounts or resources?**

A compromise of your AWS accounts or resources means that confidentiality, integrity, or availability of your data or AWS infrastructure is damaged in some way.

The likelihood of compromise indicates how likely it is that the threat scenario will result in a disruption or breach of your AWS services or resources.

As an example, consider the following configuration weaknesses:

- IAM user access keys are not rotated every 90 days.
- IAM root access key exists.

Both weaknesses are equally difficult for an adversary to take advantage of. In both cases, the adversary can use credential theft or some other method to acquire a user key. They can then use it to access your resources in an unauthorized way.

However, the likelihood of a compromise is much higher if the threat actor acquires the root user access key, because the root key gives them greater access. As a result, the root user key weakness has a higher severity.

The severity does not take into account the criticality of the underlying resource. Criticality is the level of importance of the resources that are associated with the finding. For example, a resource that is associated with a mission critical application is more critical than one that is associated with nonproduction testing. To capture resource criticality information, use the `Criticality` field of the AWS Security Finding Format (ASFF).

The following table maps the difficulty to exploit and the likelihood of compromise to the security labels.

	Compromise highly likely	Compromise likely	Compromise unlikely	Compromise highly unlikely
Very easy to exploit	Critical	Critical	High	Medium
Somewhat easy to exploit	Critical	High	Medium	Medium
Somewhat difficult to exploit	High	Medium	Medium	Low
Very difficult to exploit	Medium	Medium	Low	Low

Severity definitions

The severity labels are defined as follows.

Critical – The issue should be remediated immediately to avoid it escalating.

For example, an open S3 bucket is considered a critical severity finding. Because so many actors scan for open S3 buckets, data in exposed S3 buckets is likely to be discovered and accessed by others.

In general, resources that are publicly accessible are considered critical security issues. You should treat critical findings with the utmost urgency. You also should consider the criticality of the resource.

High – The issue must be addressed as a near-term priority.

For example, if a default VPC security group is open to inbound and outbound traffic, it is considered high severity. It is somewhat easy for a threat actor to compromise a VPC using this method. It is also likely that the threat actor will be able to disrupt or exfiltrate resources once they are in the VPC.

Security Hub recommends that you treat a high severity finding as a near-term priority. You should take immediate remediation steps. You also should consider the criticality of the resource.

Medium – The issue should be addressed as a mid-term priority.

For example, lack of encryption for data in transit is considered a medium severity finding. It requires a sophisticated man-in-the-middle attack to take advantage of this weakness. In other words, it is somewhat difficult. It is likely that some data will be compromised if the threat scenario is successful.

Security Hub recommends that you investigate the implicated resource at your earliest convenience. You also should consider the criticality of the resource.

Low – The issue does not require action on its own.

For example, failure to collect forensics information is considered low severity. This control can help to prevent future compromises, but the absence of forensics does not lead directly to a compromise.

You do not need to take immediate action on low severity findings, but they can provide context when you correlate them with other issues.

Informational – No configuration weakness was found.

In other words, the status is `PASSED`, `WARNING`, or `NOT AVAILABLE`.

There is no recommended action. Informational findings help customers to demonstrate that they are in a compliant state.

Rules for updating control findings

A subsequent check against a given rule might generate a new result. For example, the status of "Avoid the use of the root account" could change from `FAILED` to `PASSED`. In that case, a new finding is generated that contains the most recent result.

If a subsequent check against a given rule generates a result that is identical to the current result, the existing finding is updated. No new finding is generated.

Security Hub automatically archives findings from controls if the associated resource is deleted, the resource does not exist, or the control is disabled. A resource might no longer exist because the associated service is not currently used. The findings are archived automatically based on one of the following criteria:

- The finding is not updated for three to five days (note that this is best effort and not guaranteed).
- The associated AWS Config evaluation returned `NOT_APPLICABLE`.

Determining the overall status of a control from its findings

Security Hub uses the `Compliance.Status` value from each control's findings to determine the overall control status. The overall status is displayed in the control list for a standard and on the control details page.

For administrator accounts, the control status reflects the aggregated status across both the administrator account and all of the member accounts. If you have set an aggregation Region, control statuses in the aggregation Region reflect control statuses across all of your linked Regions. Specifically, the overall status of a control appears as **Failed** if the control has one or more failed findings in at least one account and one linked Region.

Security Hub typically generates the initial control status within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Statuses are only available for controls that are enabled when you visit those pages. Use the [UpdateStandardsControl](#) API operation to enable or disable a control. In addition, AWS Config resource recording must be configured for the control status to appear. After control statuses are generated for the first time, Security Hub updates the control status every 24 hours based on the findings from the previous 24 hours. On the control details page, Security Hub displays a timestamp to indicate when the status of a control was last updated.

Note

It can take up to 24 hours after enabling a control for first-time control statuses to be generated in the China Regions and AWS GovCloud (US) Region.

Values for Compliance.Status

The `Compliance.Status` for each finding is assigned one of the following values.

- `PASSED` – Automatically sets the Security Hub `Workflow.Status` to `RESOLVED`.

If `Compliance.Status` for a finding changes from `PASSED` to `FAILED`, `WARNING`, or `NOT_AVAILABLE`; and `Workflow.Status` was either `NOTIFIED` or `RESOLVED`; then Security Hub automatically sets `Workflow.Status` to `NEW`.

- **FAILED** – Indicates that the control did not pass the security check for this finding.
- **WARNING** – Indicates that the check was completed, but Security Hub cannot determine whether the resource is in a **PASSED** or **FAILED** state.
- **NOT_AVAILABLE** – Indicates that the check cannot be completed because a server failed, the resource was deleted, or the result of the AWS Config evaluation was **NOT_APPLICABLE**.

If the AWS Config evaluation result was **NOT_APPLICABLE**, then Security Hub automatically archives the finding.

Values for the control status

Security Hub uses the compliance status of the control findings to calculate an overall control status. When it calculates the overall control status, Security Hub ignores findings that have a `Workflow.Status` of **SUPPRESSED**.

The available values for the overall control status are as follows:

- **Passed** – Indicates that all findings have a `Compliance.Status` of **PASSED**.
- **Failed** – Indicates that at least one finding has a `Compliance.Status` of **FAILED**.
- **Unknown** – Indicates that at least one finding has a `Compliance.Status` of **WARNING** or **NOT_AVAILABLE**. No findings are **FAILED**.
- **No data** – Indicates that there are no findings for the control. For example, a new control has this status until it begins to generate findings. A control also has this status if all of the findings are **SUPPRESSED**.
- **Disabled** – Indicates that the control is deactivated in the current account and Region. This means that no security checks are currently being performed for this control in this account and Region.

Determining the security score for a security standard

On the **Security standards** page, Security Hub displays a security score from 0–100% for each enabled standard. The **Summary** page also displays the overall security score across all enabled standards.

When you enable Security Hub, Security Hub calculates the initial security score for a standard within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Scores are only generated for standards that are enabled when you visit those pages. To view a list of standards that are currently enabled, use the `GetEnabledStandards` API operation. In addition, AWS Config resource recording must be configured for scores to appear. The overall security score is the average of the standard security scores.

After first-time score generation, Security Hub updates the security score every 24 hours. Security Hub displays a timestamp to indicate when a security score was last updated.

Note

It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region.

How security scores are calculated

Security scores represent the proportion of **Passed** controls to enabled controls. The score is displayed as a percentage. For example, if 10 controls are enabled for a standard, and seven of those controls are in a **Passed** state, then the security score for the standard is 70%.

If your account is an administrator account, security scores account for control findings in all member accounts.

If you have set an aggregation Region, the overall security score is an aggregated score that accounts for findings in all linked Regions. Similarly, the security score for each standard is an aggregated score that accounts for findings associated with that standard in all linked Regions. Note that if your account is an administrator account, the security scores account for all member accounts and all Regions.

Security score calculation omits enabled controls that do not have any findings (overall status is **No data**). For example, a standard has 12 controls enabled. Six of those controls are in a **Passed** state. Two controls have no data. Because the calculation omits the controls without data, the security score is 60%.

Security scores for administrator accounts

For the administrator account, the overall security score and security scores for specific standards are aggregated scores across both the administrator account and all of the member accounts.

Security scores if you have set an aggregation Region

If you have set an aggregation Region, the overall security score and the security scores for each standard reflect findings from all linked Regions. If your account is an administrator account, the security scores also account for all member accounts.

Security scores on the Summary page

On the **Summary** page, the **Security standards** card displays the security scores for each enabled standard. It also displays a consolidated security score that represents the proportion of passed controls to enabled controls across all of the enabled standards.

Viewing and managing security standards

Security standards provide a set of related controls to determine compliance with regulatory frameworks, industry best practices, or company policies.

For a standard in Security Hub, you can view the list of controls and determine whether to enable or disable a standard for your account. You can also see the overall security score for the standard.

Administrator accounts see aggregated security scores and control statuses across their member accounts. If you are viewing standards from an aggregation Region, your security scores for standards incorporate the compliance statuses of controls across all of your linked Regions.

For the list of available standards and their controls, see [the section called “Available standards” \(p. 346\)](#).

For information about how to manage individual controls in a standard, see [the section called “Viewing and managing controls” \(p. 339\)](#).

Topics

- [Disabling or enabling a security standard \(p. 333\)](#)
- [Viewing details for a standard \(p. 336\)](#)

Disabling or enabling a security standard

You can disable or enable each supported security standard in Security Hub. Some security standards are enabled by default, but you can opt out of auto-enabled standards.

Security Hub is Regional. When you enable or disable a security standard, it is enabled or disabled only in the current Region or in the Regions that you specify.

When you disable a security standard, the following occurs:

- The checks for its controls are no longer performed.
- No additional findings are generated for its controls.
- Existing findings are archived automatically after three to five days (note that this is best effort and not guaranteed).
- The related AWS Config rules that Security Hub created are removed.

This normally occurs within a few minutes after you disable the standard, but might take longer.

If the first request to delete the AWS Config rules fails, then Security Hub retries every 12 hours. However, if you disabled Security Hub or you do not have any other standards enabled, then Security Hub cannot retry the request, meaning that it cannot delete the AWS Config rules. If this occurs, and you need to have AWS Config rules removed, contact AWS Support.

Before you enable any security standards, make sure that you have enabled AWS Config and configured resource recording. See [the section called “Enabling AWS Config” \(p. 8\)](#).

When you enable a security standard, all of the controls for that standard are enabled by default. You can then disable individual controls. See [the section called “Disabling and enabling individual controls” \(p. 341\)](#).

When you enable Security Hub, Security Hub calculates the initial security score for a standard within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Scores are only generated for standards that are enabled when you visit those pages. In addition, AWS Config resource recording must be configured for scores to appear. After first-time score generation, Security Hub updates the security score every 24 hours. Security Hub displays a timestamp to indicate when a security score was last updated. To view a list of standards that are currently enabled, use the [GetEnabledStandards](#) API operation.

Note

It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region.

Auto-enabled standards for new organization accounts

If you are using the integration with AWS Organizations, Security Hub automatically enables default security standards for new member accounts.

Currently, the default security standards that are automatically enabled are the AWS Foundational Security Best Practices (FSBP) standard and the Center for Internet Security (CIS) AWS Foundations Benchmark standard. For more information, see [Available security standards in AWS Security Hub](#).

Note

In order to auto-enable standards, you must first automatically enable Security Hub for new member accounts in your organization. If you turn off automatic enablement of Security Hub in new member accounts, Security Hub automatically turns off automatic enablement of default standards in new member accounts. See [Automatically enabling new organization accounts \(p. 99\)](#).

Opt out of auto-enabled standards

If you do not want to auto-enable the default security standards on new member accounts, follow these steps.

To opt out of auto-enabled standards (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. On the Security Hub navigation bar, choose **Settings**.
3. On the **Accounts** tab, turn off **Auto-enable standards**.

To opt out of auto-enabled standards (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [UpdateOrganizationConfiguration](#) operation from your Security Hub administrator account. The default value for the `AutoEnabledStandards` parameter is equal to `DEFAULT`. To opt out of auto-enabled standards in new member accounts, set `AutoEnableStandards` equal to `NONE`.
- **AWS CLI** – At the command line, run the `update-organization-configuration` command.

```
aws securityhub update-organization-configuration --auto-enable-standards
```

Disabling a security standard

On the **Security standards** page, each enabled standard includes an option to disable the standard. Follow these steps to disable a standard.

To disable a security standard (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Confirm that you are using Security Hub in the Region in which you want to disable the standard.
3. In the Security Hub navigation pane, choose **Security standards**.
4. For the standard you want to disable, choose **Disable**.

To disable a security standard (Security Hub API, AWS CLI)

- **Security Hub API** – Use the [BatchDisableStandards](#) operation. For each standard to disable, you provide the ARN of your subscription to the standard. To get the subscription ARNs for your enabled standards, use the [GetEnabledStandards](#) operation.
- **AWS CLI** – At the command line, run the `batch-disable-standards` command.

```
aws securityhub batch-disable-standards --standards-subscription-arns <subscription ARN>
```

Example

```
aws securityhub batch-disable-standards --standards-subscription-arns  
"arn:aws:securityhub:us-west-1:123456789012:subscription/aws-foundational-security-best-  
practices/v/1.0.0"
```

Enabling a security standard

On the **Security standards** page, each disabled standard includes an option to enable the standard. Follow these steps to enable a standard.

To enable a security standard (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.

2. Confirm that you are using Security Hub in the Region in which you want to enable the standard.
3. In the Security Hub navigation pane, choose **Security standards**.
4. For the standard you want to enable, choose **Enable**.

To enable a security standard (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `BatchEnableStandards` operation. To identify a standard to enable, you must provide the standard ARN. To obtain the standard ARN, use the `DescribeStandards` operation.
- **AWS CLI** – At the command line, run the `batch-enable-standards` command.

```
aws securityhub batch-enable-standards --standards-subscription-requests  
'{"StandardsArn": "<standard ARN>"}'
```

Example

```
aws securityhub batch-enable-standards --standards-subscription-requests  
'{"StandardsArn": "arn:aws:securityhub:us-east-1::standards/aws-foundational-security-  
best-practices/v/1.0.0"}'
```

Viewing details for a standard

The details page for a standard contains the list of controls in the standard. It also shows the overall score for the standard.

You can view and filter the list of controls, and perform the following actions:

- [Enable or disable a control \(p. 341\)](#)
- [View the details for a control \(p. 339\)](#). The control details page includes the list of findings for the control. See [the section called “Viewing and taking action on control findings” \(p. 343\)](#).

Displaying the details page for an enabled standard (console)

From the **Security standards** page, you can display a details page for the standard. You can only display details for an enabled standard. You cannot display details for a disabled standard.

This also applies to administrator accounts. Administrator accounts that do not have a standard enabled for their individual account cannot view the details for that standard. They cannot see the aggregated security score and status information across the member accounts that do have the standard enabled.

To display the list of controls for an enabled standard (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the Security Hub navigation pane, choose **Security standards**.
3. For the standard that you want to display the details for, choose **View results**.

Information on the standard details page

At the top of the details page is the overall score for the standard. The score is the percentage of passed controls relative to the number of enabled controls for the standard that have data. Security Hub typically calculates the initial security score within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Scores are only generated for standards that

are enabled when you visit those pages. To view a list of standards that are currently enabled, use the [GetEnabledStandards](#) API operation. In addition, AWS Config resource recording must be configured for scores to appear. After first-time score generation, Security Hub updates the security score every 24 hours. Security Hub displays a timestamp to indicate when a security score was last updated. See [the section called "Determining the standard security score" \(p. 332\)](#).

Note

It can take up to 24 hours for first-time security scores to be generated in the China Regions and AWS GovCloud (US) Region.

Next to the overall score is a chart that summarizes the control statuses. The chart shows the percentage of failed and passed controls. When you pause on the chart, the pop-up displays the following:

- The number of failed controls for each severity
- The number of controls with a status of **Unknown**
- The number of passed controls

At the bottom of the details page is the list of controls for the standard. The control list is organized and sorted based on the current overall status of the control and the severity assigned to each control. Security Hub updates the control statuses every 24 hours. A timestamp on each tab indicates when the control statuses were most recently updated. See [the section called "Determining the control status" \(p. 331\)](#).

For administrator accounts, the score and statuses are aggregated across the administrator account and all member accounts.

All of the data on the **Security standards** details pages is specific to the current Region unless you have set an aggregation Region. If you have set an aggregation Region, the security scores are cross-regional, accounting for findings in all linked Regions. The compliance status of controls on the standards details pages also reflect findings from linked Regions, and the number of security checks includes findings from linked Regions.

Filtering and sorting the controls

The control list for a standard uses tabs to provide built-in filtering for the list based on the control status. You can also filter the list based on the ID, title, and severity.

The **All enabled** tab lists all of the enabled controls for the standard. For administrator accounts, **All enabled** contains controls that are enabled in either their account or in any member account.

On the **Failed**, **Unknown**, **No data**, and **Passed** tabs, the controls from the **All enabled** tab are filtered to only include controls with that status.

The **Disabled** tab contains the list of disabled controls. For administrator accounts, the **Disabled** tab lists controls that are not enabled in either their account or any of their member accounts.

For standalone accounts and member accounts, the lists of enabled and disabled controls are updated in real time to reflect which controls are enabled and disabled. The overall status and the number of passed and failed checks for each control are updated every 24 hours.

For administrator accounts, all of the information, including the lists, is updated every 24 hours.

For each control, the control list contains the following information:

- The overall status of the control (see [the section called "Determining the control status" \(p. 331\)](#))
- The severity assigned to the control
- The control identifier and title

- The number of failed active findings and the total number of active findings. If applicable, the **Failed checks** column also lists the number of findings with a status of **Unknown**.

In addition to the built-in filters on each tab, you can filter the lists using values from the following fields:

- **Status**
- **Severity**
- **ID**
- **Title**

You can sort each list using any of the columns. By default, the **All enabled** tab is sorted so that failed controls are at the top of the list. This helps you to immediately focus on issues that require remediation.

Within each status, and on the remaining tabs, the controls are sorted by default in descending order by severity. In other words, critical controls are first, followed by high, then medium, then low severity controls.

Additional tabs for administrator accounts

For an administrator account, the lists on the first six tabs contain aggregated information across both the administrator account and their member accounts. For example, a control is listed on the **All enabled** tab if at least one of the accounts has the control enabled. A control is listed on the **Disabled** tab only if none of the accounts has the control enabled.

All of the information on these tabs is updated every 24 hours.

Administrator accounts also see the following additional tabs:

- **Enabled controls for this account** lists the controls that are enabled for the administrator account.
- **Disabled controls for this account** lists the controls that are disabled for the administrator account.

These lists are updated in real time to reflect the controls that the administrator account has enabled or disabled.

Downloading the control list

You can download the current page of the control list to a `.csv` file.

If you filtered the control list, then the downloaded file only includes the controls that match the filter.

If you chose a specific control from the list, then the downloaded file only includes that control.

To download the current page of the control list or the currently selected control, choose **Download**.

Viewing the controls for an enabled standard (Security Hub API, AWS CLI)

To display information about the controls for an enabled standard, you can use an API call or the AWS Command Line Interface.

To display the controls for an enabled standard (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DescribeStandardsControls` operation. To identify the standard to display the controls for, you provide the ARN of your subscription to the control. To get the subscription ARNs for your enabled standards, use the `GetEnabledStandards` operation.

- **AWS CLI** – At the command line, run the `describe-standards-controls` command.

```
aws securityhub describe-standards-controls --standards-subscription-arn <subscription ARN>
```

Example

```
aws securityhub describe-standards-controls --standards-subscription-arn "arn:aws:securityhub:us-east-1:123456789012:subscription/aws-foundational-security-best-practices/v/1.0.0"
```

Viewing and managing controls

A control is a security check against a specific resource. From Security Hub, you can view the control details. The control details include the control status and the findings generated for each control.

You can enable or disable a control for your account. You can also view details for and take action on the findings generated by the control.

For administrator accounts, the control details show the status and findings across all of the member accounts. If you have set an aggregation Region, the control details reflect findings across all linked Regions.

Topics

- [Viewing details for a control \(p. 339\)](#)
- [Enabling new controls automatically \(p. 340\)](#)
- [Disabling and enabling individual controls \(p. 341\)](#)
- [Viewing and taking action on control findings \(p. 343\)](#)

Viewing details for a control

For each control, you can display a page of useful details.

To display details for a control

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. [Display the controls for an enabled standard. \(p. 336\)](#)
3. From the controls list, choose the control name.

The top of the control details page provides an overview of the control, including its enablement status and the overall control status.

- **Enablement status** – The enablement status tells you whether the control is currently activated. If you are logged into an administrator account, the status displays as **Enabled** if the control is activated in at least one member account. If you have set an aggregation Region, the status displays as **Enabled** if the control is activated in at least one Region.
- **Control status** – The control status summarizes the performance of a control based on the compliance status of the control findings. Security Hub typically generates the initial control status within 30 minutes after your first visit to the **Summary** page or **Security standards** page on the Security Hub console. Statuses are only available for controls that are enabled when you visit those pages. Use the [UpdateStandardsControl](#) API operation to enable or disable a control. In addition, AWS Config

resource recording must be configured for the control status to appear. After control statuses are generated for the first time, Security Hub updates the control status every 24 hours based on the findings from the previous 24 hours. On the standard details page and the control details page, Security Hub displays a timestamp to indicate when the status was last updated. Administrator accounts see an aggregated control status across the administrator account and member accounts. If you have set an aggregation Region, the control status accounts for findings across all linked Regions. For more information about control status, see [the section called “Determining the control status” \(p. 331\)](#).

Note

It can take up to 24 hours after enabling a control for first-time control statuses to be generated in the China Regions and AWS GovCloud (US) Region.

For controls in the Payment Card Industry Data Security Standard (PCI DSS) and the CIS AWS Foundations standard, the **Related requirements** tab lists the related requirements from within the framework.

From the details page, you can disable or enable the control. See [the section called “Disabling and enabling individual controls” \(p. 341\)](#).

The bottom of the details page contains information about the active findings for the control. Control findings are generated by security checks against the control. The control finding list does not include archived findings.

The finding list uses tabs that display different subsets of the list. On most of the tabs, the finding list shows findings that have a workflow status of **NEW**, **NOTIFIED**, or **RESOLVED**. A separate tab displays **SUPPRESSED** findings.

For each finding, the list provides access to its compliance status and details about the finding and the associated resource. You can also set the workflow status of each finding and send findings to custom actions.

See [the section called “Viewing and taking action on control findings” \(p. 343\)](#).

Enabling new controls automatically

AWS Security Hub regularly adds new controls to standards. When you first enable Security Hub, it automatically enables new controls as they are added. This only applies to enabled standards. Security Hub does not enable new controls when they are added to a standard that you disabled.

You can choose whether to automatically enable new controls. If you do not automatically enable new controls, then you must enable them manually. See [the section called “Disabling and enabling individual controls” \(p. 341\)](#).

Choosing whether to automatically enable new controls (console)

The **General** tab of the **Settings** page includes a setting to control whether to automatically enable new controls.

To choose whether to enable new controls for enabled standards

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**, and then choose the **General** tab.
3. Under **Auto-enable new controls**, choose **Edit**.

4. Toggle **Auto-enable new controls in standards I have enabled**.
5. Choose **Save**.

Choosing whether to automatically enable new controls (Security Hub API, AWS CLI)

To configure whether to automatically enable new controls, you can use an API call or the AWS Command Line Interface.

To configure whether to automatically enable new controls (Security Hub API, AWS CLI)

- **Security Hub** – Use the `UpdateSecurityHubConfiguration` operation. To automatically enable controls, set `AutoEnableControls` to `true`. To not automatically enable controls, set `AutoEnableControls` to `false`.
- **AWS CLI** – At the command line, run the `update-security-hub-configuration` command. To automatically enable new controls, specify `--auto-enable-controls`. To not enable new controls, specify `--no-auto-enable-controls`.

```
aws securityhub update-security-hub-configuration --auto-enable-controls | --no-auto-enable-controls
```

Example

```
aws securityhub update-security-hub-configuration --auto-enable-controls
```

Disabling and enabling individual controls

When you enable a standard, all of the controls for that standard are enabled by default. You can then disable and enable specific controls within an enabled standard.

When you disable a control, the following occurs:

- The check for the control is no longer performed.
- No additional findings are generated for that control.
- Existing findings are archived automatically after three to five days (note that this is best effort and not guaranteed).
- The related AWS Config rules that Security Hub created are removed.

It can be useful to turn off security checks for controls that are not relevant to your environment. For example, you might prefer to use Amazon GuardDuty instead of CloudWatch alarms to monitor for anomalous activity associated with your AWS CloudTrail logs. You can then disable the CIS AWS Foundations Benchmark controls 3.1-3.14, which focus on CloudWatch alarms.

Disabling irrelevant controls reduces the number of irrelevant findings. It also removes the failed check from the security score for the associated standard.

Remember that Security Hub is Regional. When you disable or enable a control, it is disabled only in the current Region or in the Region that you specify in the API request.

Also, when you disable an entire standard, Security Hub does not track which controls were disabled. If you subsequently enable the standard again, all of the controls are enabled. For more information, see [the section called “Disabling or enabling a security standard” \(p. 333\)](#).

Note

You enable and disable controls on a region-by-region basis via the Security Hub console, API, or CLI. If you have set an aggregation Region, you see controls from all linked Regions. If a control is available in a linked Region but not in the aggregation Region, you cannot enable or disable that control from the aggregation Region. For multi-account and multi-Region control disablement scripts, refer to [Disabling Security Hub Controls in a multi-account environment](#)

Disabling a control (console)

From the Security Hub console, you can disable controls from the control list on the standard details page or from the control details page.

To disable a control (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Confirm that you are using Security Hub in the Region in which you want to disable the control.
3. In the Security Hub navigation pane, choose **Security standards**.
4. For the standard that you want to disable a control for, choose **View results**.
5. If you are an administrator account, choose **Enabled for this account**.

Other accounts can enable controls from any tab other than the **Disabled** tab.

6. Do one of the following:
 - In the control list, choose the control to disable. Then choose **Disable**.
 - Choose the control title. Then on the control details page, choose **Disable**.
7. Enter a reason why you are disabling the control. This can help others in your organization understand why the control is disabled.
8. Choose **Disable**.

Disabling a control (Security Hub API, AWS CLI)

To disable a control, you can use an API call or the AWS Command Line Interface.

To disable a control (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `UpdateStandardsControl` operation. To identify the control to disable, you provide the control ARN. To retrieve the ARNs for the controls in a standard, use the `DescribeStandardsControls` operation.
- **AWS CLI** – At the command line, run the `update-standards-control` command.

```
aws securityhub update-standards-control --standards-control-arn <control ARN> --control-status "DISABLED" --disabled-reason <description of reason to disable>
```

Example

```
aws securityhub update-standards-control --standards-control-arn "arn:aws:securityhub:us-east-1:123456789012:control/aws-foundational-security-best-practices/v/1.0.0/ACM.1" --control-status "DISABLED" --disabled-reason "Not applicable for my service"
```

Enabling a control (console)

On the standard details page, the disabled controls are displayed on the **Disabled** tab.

For administrator accounts, the **Disabled** tab contains an aggregated list across accounts. The disabled controls for the individual administrator account are displayed on the **Disabled for this account** tab.

You can enable a control from the controls list on the **Disabled** tab, or from the control details page.

To enable a disabled control (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Confirm that you are using Security Hub in the Region in which you want to disable the control.
3. In the Security Hub navigation pane, choose **Security standards**.
4. For the standard that you want to enable the control for, choose **View results**.
5. Display the list of disabled controls.

For a member account or a standalone account, choose **Disabled**.

For an administrator account, choose **Disabled for this account**.

6. Do one of the following:
 - In the control list on the **Disabled** or **Disabled for this account** tab, choose the control to enable. Then choose **Enable**.
 - Choose a control title. Then on the control details page, choose **Enable**.

Enabling a control (Security Hub API, AWS CLI)

To enable a control, you can use an API call or the AWS Command Line Interface.

To enable a control (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `UpdateStandardsControl` operation. To identify the control to enable, you provide the control ARN. To retrieve the ARNs for the controls in a standard, use the `DescribeStandardsControls` operation.
- **AWS CLI** – At the command line, run the `update-standards-control` command.

```
aws securityhub update-standards-control--standards-control-arn <control ARN> --control-status "ENABLED"
```

Example

```
aws securityhub update-standards-control --standards-control-arn "arn:aws:securityhub:us-east-1:123456789012:control/aws-foundational-security-best-practices/v/1.0.0/ACM.1" --control-status "ENABLED"
```

Viewing and taking action on control findings

The control details page contains a list of active findings for the control. The list does not include archived findings.

The control details page does not support cross-Region aggregation. The control details page only displays findings from the current Region.

The list provides tools to filter and sort the findings, so that you can focus on more urgent findings first. Each finding can include links to resource details in the related service console. For controls that are based on AWS Config rules, you can view details about the rule and the configuration timeline.

You can also use the AWS Security Hub API to retrieve a list of findings. See [the section called “Retrieving finding details \(Security Hub API, AWS CLI\)”](#) (p. 129).

Topics

- [Filtering, sorting, and downloading the control finding list](#) (p. 344)
- [Viewing details about a control finding and finding resource](#) (p. 344)
- [Taking action on control findings](#) (p. 346)

Filtering, sorting, and downloading the control finding list

The control finding list uses tabs to provide built-in filtering for the list based on the finding status. You can also filter the list based on other finding values, and download findings from the list.

Filtering and sorting the control finding list

The **All checks** tab lists all active findings that have a workflow status of `NEW`, `NOTIFIED`, or `RESOLVED`. By default, the list is sorted so that failed findings are at the top of the list. This sort order calls attention to findings that need to be addressed.

The lists on the **Failed**, **Unknown**, and **Passed** tabs are filtered based on the value of `Compliance.Status`. The lists also only include active findings that have a workflow status of `NEW`, `NOTIFIED`, or `RESOLVED`.

The **Suppressed** tab contains a list of active findings that have a workflow status of `SUPPRESSED`.

In addition to the built-in filters on each tab, you can filter the lists using values from the following fields:

- Account ID
- Workflow status
- Compliance status
- Resource ID
- Resource type

You can sort each list using any of the columns.

Downloading the control finding list

From the control details page, you can download the current page of control findings to a `.csv` file.

If you filtered the finding list, then the download only includes the controls that match the filter.

If you selected specific findings from the list, then the download only includes the selected findings.

To download the current page of the list or the selected findings, choose **Download**.

Viewing details about a control finding and finding resource

For each finding, AWS Security Hub provides access to details to help you investigate the finding.

You can display details about the finding resource and the related configuration rule.

You can also view any notes added to the finding.

Viewing the complete .json for a finding

You can display and download the full .json of a finding.

To display the .json, in the **Finding .json** column, choose the icon.

On the **Finding JSON** panel, to download the .json, choose **Download**.

Viewing information about a finding resource

The **Resource** column contains the resource type and resource identifier.

To display information about the resource, choose the resource identifier. For AWS accounts, if the account is an organization member account, then the information includes both the account ID and the account name. For accounts that were invited manually, the information only includes the account ID.

If you have permission to view the resource in its original service, then the resource identifier displays a link to the service. For example, for an AWS user, the resource details provide a link to the view the user details in IAM.

Security Hub displays a message to notify you if the resource is in a different account.

Viewing the configuration timeline for a finding resource

One avenue of investigation is the configuration timeline for the resource in AWS Config.

If you have permission to view the configuration timeline for the finding resource, then the finding list provides a link to the timeline.

Security Hub displays a message to notify you if the resource is in a different account.

To navigate to the configuration timeline in AWS Config

1. In the **Investigate** column, choose the icon.
2. On the menu, choose **Configuration timeline**. If you do not have access to the configuration timeline, then the link does not appear.

Viewing the AWS Config rule for a finding resource

If the control is based on an AWS Config rule, then you might also want to view the details for the AWS Config rule. The AWS Config rule information can help you to get a better understanding why a check passed or failed.

If you have permission to view the AWS Config rule for the control, then the finding list provides a link to the AWS Config rule in AWS Config.

Security Hub displays a message to notify you if the resource is in a different account.

To navigate to the AWS Config rule

1. In the **Investigate** column, choose the icon.
2. On the menu, choose **Config rule**. If you do not have access to the AWS Config rule, then **Config rule** is not linked.

Viewing notes for findings

If a finding has an associated note, then the **Updated** column displays a note icon.

To display the note that is associated with a finding

In the **Updated** column, choose the note icon.

Taking action on control findings

To reflect the current status of your investigation, you set the workflow status. See [the section called "Setting the workflow status for findings"](#) (p. 130).

AWS Security Hub also allows you to send selected findings to a custom action in Amazon EventBridge. See [the section called "Sending findings to a custom action"](#) (p. 131).

Available security standards in AWS Security Hub

Security standards provide a set of related controls to determine compliance with regulatory frameworks, industry best practices, or company policies.

The information for each standard includes the list of controls in the standard. Each control includes the following information:

- The security category that the control belongs to
- The resource that the control applies to
- If applicable, the AWS Config rule that is used for the control
- Any parameters used by the control
- A description of the control and what it checks
- For standards that are associated with a regulatory framework, the applicable requirements in that framework
- Information on how to remediate a failed check. For example, you might need to change the configuration of a resource.

AWS Security Hub supports the security standards listed below. If you are using the integration with AWS Organizations, the CIS and Foundational Security Best Practices standards are auto-enabled in new member accounts by default. For more information about auto-enabled standards, see [Auto-enabled standards for new organization accounts](#) (p. 334).

Topics

- [CIS AWS Foundations Benchmark standard](#) (p. 346)
- [Payment Card Industry Data Security Standard \(PCI DSS\)](#) (p. 398)
- [AWS Foundational Security Best Practices standard](#) (p. 457)

CIS AWS Foundations Benchmark standard

Security Hub supports the Center for Internet Security (CIS) AWS Foundations Benchmark standard. For more information, see [Securing Amazon Web Services](#) on the CIS website.

AWS Security Hub has satisfied the requirements of CIS Security Software Certification and has been awarded CIS Security Software Certification for the following CIS Benchmarks:

- CIS Benchmark for CIS Amazon Web Services Foundations Benchmark, v1.2.0, Level 1
- CIS Benchmark for CIS Amazon Web Services Foundations Benchmark, v1.2.0, Level 2

Topics

- [AWS Config resources required for CIS controls \(p. 347\)](#)
- [CIS AWS Foundations Benchmark controls \(p. 347\)](#)
- [CIS AWS Foundations Benchmark controls that you might want to disable \(p. 397\)](#)
- [CIS AWS Foundations Benchmark security checks that are not supported in Security Hub \(p. 398\)](#)

AWS Config resources required for CIS controls

To run security checks for the enabled controls on your environment's resources, Security Hub either runs through the exact audit steps prescribed for the checks in [Securing Amazon Web Services](#) or uses specific AWS Config managed rules.

If you don't enable all resources in AWS Config, a finding is generated for the control [2.5 – Ensure AWS Config is enabled \(p. 363\)](#). For other CIS controls, for Security Hub to accurately report findings, you must enable recording for the following resources in AWS Config.

- `AWS::CloudTrail::Trail`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::VPC`
- `AWS::IAM::Policy`
- `AWS::IAM::User`
- `AWS::KMS::Key`
- `AWS::S3::Bucket`

If a finding is generated by a security check that is based on an AWS Config rule, the finding details include a **Rules** link to open the associated AWS Config rule. To navigate to the AWS Config rule, you must also have an IAM permission in the selected account to navigate to AWS Config.

CIS AWS Foundations Benchmark controls

For the CIS AWS Foundations standard, Security Hub supports the following controls. For each control, the information includes the required AWS Config rule and the remediation steps.

1.1 – Avoid the use of the root user

Severity: Low

AWS Config rule: None

Schedule type: Periodic

The root user has unrestricted access to all services and resources in an AWS account. We highly recommend that you avoid using the root user for daily tasks. Minimizing the use of the root user and adopting the principle of least privilege for access management reduce the risk of accidental changes and unintended disclosure of highly privileged credentials.

As a best practice, use your root user credentials only when required to [perform account and service management tasks](#). Apply IAM policies directly to groups and roles but not users. For a tutorial on how to set up an administrator for daily use, see [Creating your first IAM admin user and group](#) in the *IAM User Guide*

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.3 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{$.userIdentity.type="Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent" }
```

- b. Choose **Next**.
6. Under **Assign Metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric Namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric Name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. In the navigation pane, choose **Alarms** and then **All alarms**.
9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Choose **Select metric**.
 - b. On the **Select metric** panel, scroll down to **Metrics**. Choose the **LogMetrics** namespace. You can also use the search bar to search for it.
 - c. Choose **Metrics with no dimensions**.
 - d. Select the check box for the metric that you created. Then choose **Select metric**.
 - e. Under **Metric**, leave the default values.
 - f. Under **Conditions**, for **Threshold**, choose **Static**.
 - g. For **Define the alarm condition**, choose **Greater/Equal**.
 - h. For **Define the threshold value**, enter **1**.
 - i. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm, such as **CIS-1.1-RootAccountUsage**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

1.2 – Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password

Severity: Medium

AWS Config rule: [mfa-enabled-for-iam-console-access](#)

Schedule type: Periodic

Multi-factor authentication (MFA) adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they're prompted for their user name and password as well as for an authentication code from their AWS MFA device.

CIS recommends that you enable MFA for all accounts that have a console password. MFA provides increased security for console access. It requires the authenticating principal to possess a device that emits a time-sensitive key and to have knowledge of a credential.

Important

The AWS Config rule used for this check may take up to 4 hours to accurately report results for MFA. Any findings that are generated within the first 4 hours after you enable the CIS security checks might not be accurate. It may also take up to 4 hours after you remediate this issue for the check to pass.

Remediation

To configure MFA for a user

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. Choose the **User name** of the user to configure MFA for.
4. Choose **Security credentials** and then choose **Manage** next to **Assigned MFA device**.
5. Follow the **Manage MFA Device** wizard to assign the type of device appropriate for your environment.

To learn how to delegate MFA setup to users, see [How to Delegate Management of Multi-Factor Authentication to AWS IAM Users](#) on the AWS Security Blog.

1.3 – Ensure credentials unused for 90 days or greater are disabled

Severity: Medium

AWS Config rule: [iam-user-unused-credentials-check](#)

Schedule type: Periodic

IAM users can access AWS resources using different types of credentials, such as passwords or access keys.

CIS recommends that you remove or deactivate all credentials that have been unused in 90 days or more. Disabling or removing unnecessary credentials reduces the window of opportunity for credentials associated with a compromised or abandoned account to be used.

The AWS Config rule for this control uses the [GetCredentialReport](#) and [GenerateCredentialReport](#) API operations, which are only updated every four hours. Changes to IAM users can take up to four hours to be visible to this control.

Remediation

To get some of the information that you need to monitor accounts for dated credentials, use the IAM console. For example, when you view users in your account, there is a column for **Access key age**, **Password age**, and **Last activity**. If the value in any of these columns is greater than 90 days, make the credentials for those users inactive.

You can also use credential reports to monitor user accounts and identify those with no activity for 90 or more days. You can download credential reports in .csv format from the IAM console. For more information about credential reports, see [Getting credential reports for your AWS Account](#).

After you identify the inactive accounts or unused credentials, use the following steps to disable them.

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.

3. Choose the name of the user with credentials over 90 days old.
4. Choose **Security credentials** and then choose **Make inactive** for all sign-in credentials and access keys that haven't been used in 90 days or more.

1.4 – Ensure access keys are rotated every 90 days or less

Severity: Medium

AWS Config rule: [access-keys-rotated](#)

Schedule type: Periodic

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. AWS users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services.

When you rotate access keys regularly, you reduce the chance that an access key is used that is associated with a compromised or terminated account. Rotate access keys to ensure that data can't be accessed with an old key that might have been lost, cracked, or stolen.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

To ensure that access keys aren't more than 90 days old

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. For each user that shows an **Access key age** that is greater than 90 days, choose the **User name** to open the settings for that user.
4. Choose **Security credentials**.
5. To create a new key for the user:
 - a. Choose **Create access key**.
 - b. To save the key content, either download the secret access key, or choose **Show** and then copy it from the page.
 - c. Store the key in a secure location to provide to the user.
 - d. Choose **Close**.
6. Update all applications that were using the previous key to use the new key.
7. For the previous key, choose **Make inactive** to make the access key inactive. Now the user can't make requests using that key.
8. Confirm that all applications work as expected with the new key.
9. After confirming that all applications work with the new key, delete the previous key. After you delete the access key, you can't recover it.

To delete the previous key, choose the **X** at the end of the row and then choose **Delete**.

1.5 – Ensure IAM password policy requires at least one uppercase letter

Severity: Medium

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets.

CIS recommends that the password policy require at least one uppercase letter. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.
3. Select **Requires at least one uppercase letter** and then choose **Apply password policy**.

1.6 – Ensure IAM password policy requires at least one lowercase letter

Severity: Medium

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets. CIS recommends that the password policy require at least one lowercase letter. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.
3. Select **Requires at least one lowercase letter** and then choose **Apply password policy**.

1.7 – Ensure IAM password policy requires at least one symbol

Severity: Medium

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets.

CIS recommends that the password policy require at least one symbol. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.

2. Choose **Account settings**.
3. Select **Require at least one non-alphanumeric character** and then choose **Apply password policy**.

1.8 – Ensure IAM password policy requires at least one number

Severity: Medium

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords use different character sets.

CIS recommends that the password policy require at least one number. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.
3. Select **Requires at least one number** and then choose **Apply password policy**.

1.9 – Ensure IAM password policy requires a minimum length of 14 or greater

Severity: Medium

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

Password policies, in part, enforce password complexity requirements. Use IAM password policies to ensure that passwords are at least a given length.

CIS recommends that the password policy require a minimum password length of 14 characters. Setting a password complexity policy increases account resiliency against brute force login attempts.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.
3. In the **Minimum password length** field, enter **14**, then choose **Apply password policy**.

1.10 – Ensure IAM password policy prevents password reuse

Severity: Low

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

This control checks whether the number of passwords to remember is set to 24. The control fails if the value is not 24.

IAM password policies can prevent the reuse of a given password by the same user.

CIS recommends that the password policy prevent the reuse of passwords. Preventing password reuse increases account resiliency against brute force login attempts.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.
3. Select **Prevent password reuse** and then enter **24** for **Number of passwords to remember**.
4. Choose **Apply password policy**.

1.11 – Ensure IAM password policy expires passwords within 90 days or less

Severity: Low

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

IAM password policies can require passwords to be rotated or expired after a given number of days.

CIS recommends that the password policy expire passwords after 90 days or less. Reducing the password lifetime increases account resiliency against brute force login attempts. Requiring regular password changes also helps in the following scenarios:

- Passwords can be stolen or compromised without your knowledge. This can happen via a system compromise, software vulnerability, or internal threat.
- Certain corporate and government web filters or proxy servers can intercept and record traffic even if it's encrypted.
- Many people use the same password for many systems such as work, email, and personal.
- Compromised end-user workstations might have a keystroke logger.

Remediation

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.
3. Select **Enable password expiration** and then enter **90** for **Password expiration period (in days)**.
4. Choose **Apply password policy**.

1.12 – Ensure no root user access key exists

Severity: Critical

AWS Config rule: [iam-root-access-key-check](#)

Schedule type: Periodic

The root user has complete access to all services and resources in an AWS account. AWS Access Keys provide programmatic access to a given account.

CIS recommends that all access keys be associated with the root user be removed. Removing access keys associated with the root user limits vectors that the account can be compromised by. Removing the root user access keys also encourages the creation and use of role-based accounts that are least privileged.

Note

This control is not supported in Asia Pacific (Osaka).

Remediation

To delete access keys

1. Log in to your account using the root user credentials.
2. Choose the account name near the top-right corner of the page and then choose **Security Credentials**.
3. Choose **Access keys (access key ID and secret access key)**.
4. To permanently delete the key, choose **Delete** and then choose **Yes**. You cannot recover deleted keys.
5. If there is more than one root user access key, then repeat steps 4 and 5 for each key.

1.13 – Ensure MFA is enabled for the root user

Severity: Critical

AWS Config rule: `root-account-mfa-enabled`

Schedule type: Periodic

The root user has complete access to all the services and resources in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they're prompted for their user name and password and for an authentication code from their AWS MFA device.

When you use virtual MFA for the root user, CIS recommends that the device used is *not* a personal device. Instead, use a dedicated mobile device (tablet or phone) that you manage to keep charged and secured independent of any individual personal devices. This lessens the risks of losing access to the MFA due to device loss, device trade-in, or if the individual owning the device is no longer employed at the company.

Note

This control is not supported in the following Regions.

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West).

Remediation

To enable MFA for the root user

1. Log in to your account using the root user credentials.
2. Choose the account name near the top-right corner of the page and then choose **My Security Credentials**.

3. In the pop-up warning, choose **Continue to Security Credentials**.
4. Choose **Multi-factor authentication (MFA)**.
5. Choose **Activate MFA**.
6. Choose the type of device to use for MFA and then choose **Continue**.
7. Complete the steps to configure the device type appropriate to your selection.

Choose a hardware-based authentication mechanism for best results in passing the check [1.14 – Ensure hardware MFA is enabled for the root user](#) (p. 356).

1.14 – Ensure hardware MFA is enabled for the root user

Severity: Critical

AWS Config rule: `root-account-hardware-mfa-enabled`

Schedule type: Periodic

The root user has complete access to all services and resources in an AWS account. MFA adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they're prompted for their user name and password and for an authentication code from their AWS MFA device.

For Level 2, CIS recommends that you protect root user credentials with a hardware MFA. A hardware MFA has a smaller attack surface than a virtual MFA. For example, a hardware MFA doesn't suffer the attack surface introduced by the mobile smartphone that a virtual MFA resides on.

Using hardware MFA for many, many accounts might create a logistical device management issue. If this occurs, consider implementing this Level 2 recommendation selectively to the highest security accounts. You can then apply the Level 1 recommendation to the remaining accounts.

Both time-based one-time password (TOTP) and Universal 2nd Factor (U2F) tokens are viable as hardware MFA options.

Note

This control is not supported in the following Regions.

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West).

Remediation

To enable hardware-based MFA for the root user

1. Log in to your account using the root user credentials.
2. Choose the account name near the top-right corner of the page and then choose **My Security Credentials**.
3. In the pop-up warning, choose **Continue to Security Credentials**.
4. Choose **Multi-factor authentication (MFA)**.
5. Choose **Activate MFA**.
6. Choose a hardware-based (not virtual) device to use for MFA and then choose **Continue**.
7. Complete the steps to configure the device type appropriate to your selection.

1.16 – Ensure IAM policies are attached only to groups or roles

Severity: Low

AWS Config rule: [iam-user-no-policies-check](#)

Schedule type: Change triggered

By default, IAM users, groups, and roles have no access to AWS resources. IAM policies are how privileges are granted to users, groups, or roles.

CIS recommends that you apply IAM policies directly to groups and roles but not users. Assigning privileges at the group or role level reduces the complexity of access management as the number of users grow. Reducing access management complexity might in turn reduce opportunity for a principal to inadvertently receive or retain excessive privileges.

Note

IAM users created by Amazon Simple Email Service are automatically created using inline policies. Security Hub automatically exempts these users from this control.

Remediation

To resolve this issue, create an IAM group, assign the policy to the group, and then add the users to the group. The policy is applied to each user in the group.

To create an IAM group

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Groups** and then choose **Create New Group**.
3. Enter a name for the group to create and then choose **Next Step**.
4. Select each policy to assign to the group and then choose **Next Step**.

The policies that you choose should include any policies currently attached directly to a user account. The next step to resolve a failed check is to add users to a group and then assign the policies to that group. Each user in the group gets assigned the policies assigned to the group.

5. Confirm the details on the **Review** page and then choose **Create Group**.

For more information about creating groups, see [Creating IAM groups](#) in the *IAM User Guide*.

To add users to an IAM group

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Groups**.
3. Choose **Group Actions** and then choose **Add Users to Group**.
4. Select the users to add to the group and then choose **Add Users**.

For more information about adding users to groups, see [Adding and removing users in an IAM group](#).

To remove a policy attached directly to a user

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. For the user to detach a policy from, choose the name in the **User name** column.
4. For each policy listed under **Attached directly**, choose the **X** on the right side of the page to remove the policy from the user and then choose **Remove**.

5. Confirm that the user can still use AWS services as expected.

1.20 - Ensure a support role has been created to manage incidents with AWS Support

Severity: Low

AWS Config rule: [iam-policy-in-use](#)

Schedule type: Periodic

AWS provides a support center that can be used for incident notification and response, as well as technical support and customer services.

Create an IAM role to allow authorized users to manage incidents with AWS Support. By implementing least privilege for access control, an IAM role will require an appropriate IAM policy to allow support center access in order to manage incidents with AWS Support.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Remediation

To remediate this issue, create a role to allow authorized users to manage AWS Support incidents.

To create the role to use for AWS Support access

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the IAM navigation pane, choose **Roles**, then choose **Create role**.
3. For **Role type**, choose the **Another AWS account**.
4. For **Account ID**, enter the AWS account ID of the AWS account to which you want to grant access to your resources.

If the users or groups that will assume this role are in the same account, then enter the local account number.

Note

The administrator of the specified account can grant permission to assume this role to any IAM user in that account. To do this, the administrator attaches a policy to the user or a group that grants permission for the `sts:AssumeRole` action. In that policy, the resource must be the role ARN.

5. Choose **Next: Permissions**.
6. Search for the managed policy `AWSSupportAccess`.
7. Select the check box for the `AWSSupportAccess` managed policy.
8. Choose **Next: Tags**.
9. (Optional) To add metadata to the role, attach tags as key-value pairs.

For more information about using tags in IAM, see [Tagging IAM users and roles](#) in the *IAM User Guide*.

10. Choose **Next: Review**.
11. For **Role name**, enter a name for your role.

Role names must be unique within your AWS account. They are not case sensitive.

12. (Optional) For **Role description**, enter a description for the new role.
13. Review the role, then choose **Create role**.

1.22 – Ensure IAM policies that allow full "*" administrative privileges are not created

Severity: High

AWS Config rule: `iam-policy-no-statements-with-admin-access`

Schedule type: Change triggered

IAM policies define a set of privileges granted to users, groups, or roles. It's recommended and considered a standard security advice to grant least privilege—that is, granting only the permissions required to perform a task. Determine what users need to do and then craft policies that let the users perform only those tasks, instead of allowing full administrative privileges.

It's more secure to start with a minimum set of permissions and grant additional permissions as necessary, rather than starting with permissions that are too lenient and then trying to tighten them later. Providing full administrative privileges instead of restricting to the minimum set of permissions that the user is required to do exposes the resources to potentially unwanted actions.

You should remove IAM policies that have a statement with "Effect": "Allow" with "Action": "*" over "Resource": "*".

Remediation

To modify an IAM policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Policies**.
3. Select the radio button next to the policy to remove.
4. From the **Policy actions** drop-down menu, choose **Detach**.
5. On the **Detach policy** page, select the radio button next to each user to detach the policy from and then choose **Detach policy**.

Confirm that the user that you detached the policy from can still access AWS services and resources as expected.

2.1 – Ensure CloudTrail is enabled in all Regions

Severity: High

AWS Config rule: `multi-region-cloudtrail-enabled`

Schedule type: Periodic

This control checks that there is at least one multi-Region CloudTrail trail. It also checks that the `ExcludeManagementEventSources` parameter is empty for at least one of those trails.

CloudTrail is a service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.

CloudTrail provides a history of AWS API calls for an account, including API calls made via the AWS Management Console, AWS SDKs, command-line tools, and higher-level AWS services (such as AWS CloudFormation).

The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing. Additionally:

- Ensuring that a multi-Region trail exists ensures that unexpected activity occurring in otherwise unused Regions is detected
- Ensuring that a multi-Region trail exists ensures that Global Service Logging is enabled for a trail by default to capture recording of events generated on AWS global services
- For a multi-Region trail, ensuring that management events configured for all type of Read/Writes ensures recording of management operations that are performed on all resources in an AWS account

By default, CloudTrail trails that are created using the AWS Management Console are multi-Region trails.

Remediation

To create a new trail in CloudTrail

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. If you haven't used CloudTrail before, choose **Get Started Now**.
3. Choose **Trails** and then choose **Create trail**.
4. Enter a name for the trail.
5. Under **Storage location**, do one of the following:
 - To create a new S3 bucket for CloudTrail logs, choose **Create new S3 bucket** and then enter a name for the bucket.
 - Choose **Use existing S3 bucket** and then select the bucket to use.
6. Choose **Additional settings** and, for **Log file validation**, choose **Enabled** to pass [2.2 – Ensure CloudTrail log file validation is enabled \(p. 361\)](#).
7. To pass [the section called “2.4 – Ensure CloudTrail trails are integrated with Amazon CloudWatch Logs” \(p. 362\)](#), you must enable CloudWatch Logs.
 - a. Under CloudWatch Logs, select the **Enabled** check box.
 - b. For **Log group**, do one of the following:
 - To use an existing log group, choose **Existing** and then enter the name of the log group to use.
 - To create a new log group, choose **New** and then enter a name for the log group to create.
 - c. For **IAM role**, do one of the following:
 - To use an existing role, choose **Existing** and then choose the role from the drop-down list.
 - To create a new role, choose **New** and then enter a name for the role to create. The new role is assigned a policy that grants the necessary permissions.

To view the permissions granted to the role, expand the **Policy document**.
8. Choose **Create**.

To update an existing trail in CloudTrail

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.

3. Choose the name of the trail in the **Name** column.
4. Update the trail configuration as needed.

To update the configuration in a particular section, do the following:

- a. Choose **Edit** for that section.
- b. Make the required updates to the configuration.
- c. Choose **Save changes**.

2.2. – Ensure CloudTrail log file validation is enabled

Severity: Medium

AWS Config rule: `cloud-trail-log-file-validation-enabled`

Schedule type: Periodic

CloudTrail log file validation creates a digitally signed digest file containing a hash of each log that CloudTrail writes to S3. You can use these digest files to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log.

CIS recommends that you enable file validation on all trails. Enabling log file validation provides additional integrity checking of CloudTrail logs.

Remediation

To enable CloudTrail log file validation

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Choose the name of a trail to edit in the **Name** column.
4. Under **General details**, choose **Edit**.
5. Under **Additional settings**, for **Log file validation**, select **Enabled**.
6. Choose **Save**.

2.3 – Ensure the S3 bucket CloudTrail logs to is not publicly accessible

Severity: Critical

AWS Config rules: `s3-bucket-public-read-prohibited`, `s3-bucket-public-write-prohibited`

Schedule type: Periodic and change triggered

CloudTrail logs a record of every API call made in your account. These log files are stored in an S3 bucket. CIS recommends that the S3 bucket policy, or access control list (ACL), applied to the S3 bucket that CloudTrail logs to prevents public access to the CloudTrail logs. Allowing public access to CloudTrail log content might aid an adversary in identifying weaknesses in the affected account's use or configuration.

To run this check, Security Hub first uses custom logic to look for the S3 bucket where your CloudTrail logs are stored. It then uses the AWS Config managed rules to check that bucket is publicly accessible.

If you aggregate your logs into a single centralized S3 bucket, then Security Hub only runs the check against the account and Region where the centralized S3 bucket is located. For other accounts and Regions, the control status is **No data**.

If the bucket is publicly accessible, the check generates a failed finding.

Remediation

To remove public access for an Amazon S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the name of the bucket where your CloudTrail are stored.
3. Choose **Permissions** and then choose **Public access settings**.
4. Choose **Edit**, select all four options, and then choose **Save**.
5. If prompted, enter **confirm** and then choose **Confirm**.

2.4 – Ensure CloudTrail trails are integrated with Amazon CloudWatch Logs

Severity: Low

AWS Config rule: `cloud-trail-cloud-watch-logs-enabled`

Schedule type: Periodic

CloudTrail is a web service that records AWS API calls made in a given account. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.

CloudTrail uses Amazon S3 for log file storage and delivery, so log files are stored durably. In addition to capturing CloudTrail logs in a specified Amazon S3 bucket for long-term analysis, you can perform real-time analysis by configuring CloudTrail to send logs to CloudWatch Logs.

For a trail that is enabled in all Regions in an account, CloudTrail sends log files from all those Regions to a CloudWatch Logs log group.

CIS recommends that you send CloudTrail logs to CloudWatch Logs.

Note

The intent of this recommendation is to ensure that account activity is being captured, monitored, and appropriately alarmed on. CloudWatch Logs is a native way to accomplish this using AWS services but doesn't preclude the use of an alternate solution.

Sending CloudTrail logs to CloudWatch Logs facilitates real-time and historic activity logging based on user, API, resource, and IP address. It provides the opportunity to establish alarms and notifications for anomalous or sensitivity account activity.

Remediation

To ensure that CloudTrail trails are integrated with CloudWatch Logs

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Choose a trail that there is no value for in the **CloudWatch Logs Log group** column.
4. Scroll down to the **CloudWatch Logs** section and then choose **Edit**.
5. Select the **Enabled** check box.
6. For **Log group**, do one of the following:
 - To use an existing log group, choose **Existing** and then enter the name of the log group to use.

- To create a new log group, choose **New** and then enter a name for the log group to create.
7. For **IAM role**, do one of the following:
 - To use an existing role, choose **Existing** and then choose the role from the drop-down list.
 - To create a new role, choose **New** and then enter a name for the role to create. The new role is assigned a policy that grants the necessary permissions.

To view the permissions granted to the role, expand the **Policy document**.
 8. Choose **Save changes**.

For more information, see [Configuring CloudWatch Logs monitoring with the console](#) in the *AWS CloudTrail User Guide*.

2.5 – Ensure AWS Config is enabled

Severity: Medium

AWS Config rule: None

Schedule type: Periodic

AWS Config is a web service that performs configuration management of supported AWS resources in your account and delivers log files to you. The recorded information includes the configuration item (AWS resource), relationships between configuration items (AWS resources), and any configuration changes between resources.

CIS recommends that you enable AWS Config in all Regions. The AWS configuration item history that AWS Config captures enables security analysis, resource change tracking, and compliance auditing.

Note

CIS 2.5 requires that AWS Config is enabled in all Regions in which you use Security Hub. Because Security Hub is a regional service, the check performed for this control checks only the current Region for the account. It does not check all Regions. You also must record global resources so that security checks against global resources can be checked in each Region. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

To run this check, Security Hub performs custom logic to perform the audit steps prescribed for it in the [CIS AWS Foundations Benchmark v1.2](#). Security Hub also requires that global resources are recorded in each Region, because Security Hub is a regional service and performs its security checks on a Region-by-Region basis.

Remediation

To configure AWS Config settings

1. Open the AWS Config console at <https://console.aws.amazon.com/config/>.
2. Select the Region to configure AWS Config in.
3. If you haven't used AWS Config before, see [Getting Started](#) in the *AWS Config Developer Guide*.
4. Navigate to the Settings page from the menu, and do the following:
 - Choose **Edit**.
 - Under **Resource types to record**, select **Record all resources supported in this region** and **Include global resources (e.g., AWS IAM resources)**.
 - Under **Data retention period**, choose the default retention period for AWS Config data, or specify a custom retention period.

- Under **AWS Config role**, either choose **Create AWS Config service-linked role** or choose **Choose a role from your account** and then select the role to use.
 - Under **Amazon S3 bucket**, specify the bucket to use or create a bucket and optionally include a prefix.
 - Under **Amazon SNS topic**, select an Amazon SNS topic from your account or create one. For more information about Amazon SNS, see the [Amazon Simple Notification Service Getting Started Guide](#).
5. Choose **Save**.

For more information about using AWS Config from the AWS Command Line Interface, see [Turning on AWS Config](#) in the *AWS Config Developer Guide*.

You can also use an AWS CloudFormation template to automate this process. For more information, see the [AWS CloudFormation StackSets sample template](#) in the *AWS CloudFormation User Guide*.

2.6 – Ensure S3 bucket access logging is enabled on the CloudTrail S3 bucket

Severity: Low

AWS Config rule: [s3-bucket-logging-enabled](#)

Schedule type: Periodic

Amazon S3 bucket access logging generates a log that contains access records for each request made to your S3 bucket. An access log record contains details about the request, such as the request type, the resources specified in the request worked, and the time and date the request was processed.

CIS recommends that you enable bucket access logging on the CloudTrail S3 bucket.

By enabling S3 bucket logging on target S3 buckets, you can capture all events that might affect objects in a target bucket. Configuring logs to be placed in a separate bucket enables access to log information, which can be useful in security and incident response workflows.

To run this check, Security Hub first uses custom logic to look for the bucket where your CloudTrail logs are stored and then uses the AWS Config managed rule to check if logging is enabled.

If you aggregate your logs into a single centralized S3 bucket, then Security Hub only runs the check against the account and Region where the centralized S3 bucket is located. For other accounts and Regions, the control status is **No data**.

If the bucket is publicly accessible, the check generates a failed finding.

Remediation

To enable S3 bucket access logging

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the bucket used for CloudTrail.
3. Choose **Properties**.
4. Choose **Server access logging**, then choose **Enable logging**.
5. Select a bucket from the **Target bucket** list, and optionally enter a prefix.
6. Choose **Save**.

2.7 – Ensure CloudTrail logs are encrypted at rest using AWS KMS keys

Severity: Medium

AWS Config rule: `cloud-trail-encryption-enabled`

Schedule type: Periodic

CloudTrail is a web service that records AWS API calls for an account and makes those logs available to users and resources in accordance with IAM policies. AWS Key Management Service (AWS KMS) is a managed service that helps create and control the encryption keys used to encrypt account data, and uses hardware security modules (HSMs) to protect the security of encryption keys.

You can configure CloudTrail logs to leverage server-side encryption (SSE) and KMS keys to further protect CloudTrail logs.

CIS recommends that you configure CloudTrail to use SSE-KMS.

Configuring CloudTrail to use SSE-KMS provides additional confidentiality controls on log data because a given user must have S3 read permission on the corresponding log bucket and must be granted decrypt permission by the KMS key policy.

Remediation

To enable encryption for CloudTrail logs

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Choose the trail to update.
4. Under **Storage location**, choose the pencil icon to edit the settings.
5. For **Encrypt log files with SSE-KMS**, choose **Yes**.
6. For **Create a new KMS key**, do one of the following:
 - To create a key, choose **Yes** and then enter an alias for the key in the **KMS key** field. The key is created in the same Region as the bucket.
 - To use an existing key, choose **No** and then select the key from the **KMS key** list.

Note

The AWS KMS key and S3 bucket must be in the same Region.

7. Choose **Save**.

You might need to modify the policy for CloudTrail to successfully interact with your KMS key. For more information, see [Encrypting CloudTrail log files with AWS KMS–Managed Keys \(SSE-KMS\)](#) in the *AWS CloudTrail User Guide*.

2.8 – Ensure rotation for customer-created KMS keys is enabled

Severity: Medium

AWS Config rule: `cmk-backing-key-rotation-enabled`

Schedule type: Periodic

AWS KMS enables customers to rotate the backing key, which is key material stored in AWS KMS and is tied to the key ID of the KMS key. It's the backing key that is used to perform cryptographic operations such as encryption and decryption. Automated key rotation currently retains all previous backing keys so that decryption of encrypted data can take place transparently.

CIS recommends that you enable KMS key rotation. Rotating encryption keys helps reduce the potential impact of a compromised key because data encrypted with a new key can't be accessed with a previous key that might have been exposed.

Remediation

To enable KMS key rotation

1. Open the AWS KMS console at <https://console.aws.amazon.com/kms>.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. Choose **Customer managed keys**.
4. Choose the alias of the key to update in the **Alias** column.
5. Choose **Key rotation**.
6. Select **Automatically rotate this KMS key every year** and then choose **Save**.

2.9 – Ensure VPC flow logging is enabled in all VPCs

Severity: Medium

AWS Config rule: `vpc-flow-logs-enabled`

Schedule type: Periodic

VPC flow logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. After you have created a flow log, you can view and retrieve its data in CloudWatch Logs.

CIS recommends that you enable flow logging for packet rejects for VPCs. Flow logs provide visibility into network traffic that traverses the VPC and can detect anomalous traffic or insight during security workflows.

Remediation

To enable VPC flow logging

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Choose **Your VPCs**.
3. Select a VPC to update.
4. Choose the **Flow Logs** tab in the bottom section of the page.
5. Choose **Create flow log**.
6. For **Filter**, choose **Reject**.
7. For **Destination log group**, select the log group to use.
8. For **IAM role**, select the IAM role to use.
9. Choose **Create**.

3.1 – Ensure a log metric filter and alarm exist for unauthorized API calls

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm unauthorized API calls. Monitoring unauthorized API calls helps reveal application errors and might reduce time to detect malicious activity.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.1 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region. The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:

- a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.errorCode="*UnauthorizedOperation") || ($.errorCode="AccessDenied*)}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.
9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, for **Statistic**, choose **Average**. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.1-UnauthorizedAPICalls**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.2 – Ensure a log metric filter and alarm exist for AWS Management Console sign-in without MFA

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm console logins that aren't protected by MFA. Monitoring for single-factor console logins increases visibility into accounts that aren't protected by MFA.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.2 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.

4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed != "Yes")  
&& ($.userIdentity.type = "IAMUser") && ($.responseElements.ConsoleLogin =  
"Success") }
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.2-ConsoleSigninWithoutMFA**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.3 – Ensure a log metric filter and alarm exist for usage of root user

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for root user login attempts. Monitoring for root user logins provides visibility into the use of a fully privileged account and an opportunity to reduce the use of it.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.3 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.

3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{$.userIdentity.type="Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent" }
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **RootAccountUsage**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.4 – Ensure a log metric filter and alarm exist for IAM policy changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes made to IAM policies. Monitoring these changes helps ensure that authentication and authorization controls remain intact.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.4 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

Note that the alarm checks for specific API operations by name. One of these operations is `DeletePolicy`. The alarm does not check that the call was issued from IAM. Because of this, the alarm also is triggered when Auto Scaling calls `DeletePolicy`.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventName=DeleteGroupPolicy) || ($.eventName=DeleteRolePolicy) || ($.eventName=DeleteUserPolicy) || ($.eventName=PutGroupPolicy) || ($.eventName=PutRolePolicy) || ($.eventName=PutUserPolicy) || ($.eventName=CreatePolicy) || ($.eventName=DeletePolicy) || ($.eventName=CreatePolicyVersion) || ($.eventName=DeletePolicyVersion) || ($.eventName=AttachRolePolicy) || ($.eventName=DetachRolePolicy) || ($.eventName=AttachUserPolicy) || ($.eventName=DetachUserPolicy) || ($.eventName=AttachGroupPolicy) || ($.eventName=DetachGroupPolicy)}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.
 9. Choose **Create Alarm**.
 10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, for **Statistic**, choose **Average**. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
 11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.

12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.4-IAMPolicyChanges**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.5 – Ensure a log metric filter and alarm exist for CloudTrail configuration changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes to CloudTrail configuration settings. Monitoring these changes helps ensure sustained visibility to activities in the account.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.5 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventName=CreateTrail) || ($.eventName=UpdateTrail) ||  
 ($.eventName>DeleteTrail) || ($.eventName=StartLogging) ||  
 ($.eventName=StopLogging)}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.

- c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.5-CloudTrailChanges**. Then choose **Next**.
 13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.6 – Ensure a log metric filter and alarm exist for AWS Management Console authentication failures

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for failed console authentication attempts. Monitoring failed console logins might decrease lead time to detect an attempt to brute-force a credential, which might provide an indicator, such as source IP, that you can use in other event correlations.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.6 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in **FAILED** findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of **NO_DATA** in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates **WARNING** findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.

2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventName=ConsoleLogin) && ($.errorMessage="Failed authentication")}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.
 9. Choose **Create Alarm**.
 10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
 11. Under **Configure actions**, do the following:

- a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.6-ConsoleAuthenticationFailure**. Then choose **Next**.
 13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.7 – Ensure a log metric filter and alarm exist for disabling or scheduled deletion of customer managed keys

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for customer managed keys that have changed state to disabled or scheduled deletion. Data encrypted with disabled or deleted keys is no longer accessible.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.7 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters. The control also fails if `ExcludeManagementEventSources` contains `kms.amazonaws.com`.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in **FAILED** findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of **NO_DATA** in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates **WARNING** findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions"](#) (p. 359).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventSource=kms.amazonaws.com) && (($.eventName=DisableKey) || ($.eventName=ScheduleKeyDeletion))}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.

- d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
- a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.7-DisableOrDeleteCMK**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.8 – Ensure a log metric filter and alarm exist for S3 bucket policy changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes to S3 bucket policies. Monitoring these changes might reduce time to detect and correct permissive policies on sensitive S3 buckets.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.8 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in **FAILED** findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of **NO_DATA** in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates **WARNING** findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in the section called “2.1 – Ensure CloudTrail is enabled in all Regions” (p. 359).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{ ($.eventSource=s3.amazonaws.com) && (( $.eventName=PutBucketAcl  
|| ($.eventName=PutBucketPolicy) || ($.eventName=PutBucketCors) ||  
($.eventName=PutBucketLifecycle) || ($.eventName=PutBucketReplication)  
|| ($.eventName=DeleteBucketPolicy) || ($.eventName=DeleteBucketCors) ||  
($.eventName=DeleteBucketLifecycle) || ($.eventName=DeleteBucketReplication)) }
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, for **Statistic**, choose **Average**. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.

- b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
 12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.8-S3BucketPolicyChanges**. Then choose **Next**.
 13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.9 – Ensure a log metric filter and alarm exist for AWS Config configuration changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms.

CIS recommends that you create a metric filter and alarm for changes to AWS Config configuration settings. Monitoring these changes helps ensure sustained visibility of configuration items in the account.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.9 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region. The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventSource=config.amazonaws.com) && (($.eventName=StopConfigurationRecorder) || ($.eventName>DeleteDeliveryChannel) || ($.eventName=PutDeliveryChannel) || ($.eventName=PutConfigurationRecorder))}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.

10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.9-AWSConfigChanges**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.10 – Ensure a log metric filter and alarm exist for security group changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Security groups are a stateful packet filter that controls ingress and egress traffic in a VPC.

CIS recommends that you create a metric filter and alarm for changes to security groups. Monitoring these changes helps ensure that resources and services aren't unintentionally exposed.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.10 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region. The check results in **FAILED** findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of **NO_DATA** in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{ ($.eventName=AuthorizeSecurityGroupIngress)
  || ($.eventName=AuthorizeSecurityGroupEgress)
  || ($.eventName=RevokeSecurityGroupIngress) ||
  ($.eventName=RevokeSecurityGroupEgress) || ($.eventName=CreateSecurityGroup) ||
  ($.eventName>DeleteSecurityGroup)}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
 7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.

8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.

10. Under **Specify metric and conditions**, do the following:

- a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
- b. Under **Conditions**, for **Threshold**, choose **Static**.
- c. For **Define the alarm condition**, choose **Greater/Equal**.
- d. For **Define the threshold value**, enter **1**.
- e. Choose **Next**.

11. Under **Configure actions**, do the following:

- a. Under **Alarm state trigger**, choose **In alarm**.
- b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
- c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
- d. Choose **Next**.

12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.10-SecurityGroupChanges**. Then choose **Next**.

13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.11 – Ensure a log metric filter and alarm exist for changes to Network Access Control Lists (NACL)

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. NACLs are used as a stateless packet filter to control ingress and egress traffic for subnets in a VPC.

CIS recommends that you create a metric filter and alarm for changes to NACLs. Monitoring these changes helps ensure that AWS resources and services aren't unintentionally exposed.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.11 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in **FAILED** findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of **NO_DATA** in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventName=CreateNetworkAcl) || ($.eventName=CreateNetworkAclEntry) || ($.eventName>DeleteNetworkAcl) || ($.eventName>DeleteNetworkAclEntry) || ($.eventName=ReplaceNetworkAclEntry) || ($.eventName=ReplaceNetworkAclAssociation)}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.

- d. For **Metric value**, enter **1**.
- e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.
9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.11-NetworkACLChanges**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.12 – Ensure a log metric filter and alarm exist for changes to network gateways

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Network gateways are required to send and receive traffic to a destination outside a VPC.

CIS recommends that you create a metric filter and alarm for changes to network gateways. Monitoring these changes helps ensure that all ingress and egress traffic traverses the VPC border via a controlled path.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.12 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in **FAILED** findings in the following cases:

- No trail is configured.

- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called "2.1 – Ensure CloudTrail is enabled in all Regions" \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{ ($.eventName=CreateCustomerGateway) || ($.eventName>DeleteCustomerGateway) ||  
  ($.eventName=AttachInternetGateway) || ($.eventName>CreateInternetGateway) ||  
  ($.eventName>DeleteInternetGateway) || ($.eventName=DetachInternetGateway) }
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.

- c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter 1.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
 8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.

9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, leave the default values. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter 1.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.12-NetworkGatewayChanges**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.13 – Ensure a log metric filter and alarm exist for route table changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. Routing tables route network traffic between subnets and to network gateways.

CIS recommends that you create a metric filter and alarm for changes to route tables. Monitoring these changes helps ensure that all VPC traffic flows through an expected path.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.13 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventName=CreateRoute) || ($.eventName=CreateRouteTable) ||  
($.eventName=ReplaceRoute) || ($.eventName=ReplaceRouteTableAssociation)  
|| ($.eventName>DeleteRouteTable) || ($.eventName>DeleteRoute) ||  
($.eventName=DisassociateRouteTable)}
```

- b. Choose **Next**.

6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.
9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, for **Statistic**, choose **Average**. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.13-RouteTableChanges**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

3.14 – Ensure a log metric filter and alarm exist for VPC changes

Severity: Low

AWS Config rule: None

Schedule type: Periodic

You can do real-time monitoring of API calls by directing CloudTrail logs to CloudWatch Logs and establishing corresponding metric filters and alarms. You can have more than one VPC in an account, and you can create a peer connection between two VPCs, enabling network traffic to route between VPCs.

CIS recommends that you create a metric filter and alarm for changes to VPCs. Monitoring these changes helps ensure that authentication and authorization controls remain intact.

To run this check, Security Hub uses custom logic to perform the exact audit steps prescribed for control 3.14 in the [CIS AWS Foundations Benchmark v1.2](#). This control fails if the exact metric filters prescribed by CIS are not used. Additional fields or terms cannot be added to the metric filters.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a CloudTrail trail, a metric filter, and an alarm for the metric filter.

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic. For more information, see [Getting started with Amazon SNS](#) in the *Amazon Simple Notification Service Developer Guide*.

Next, set up an active CloudTrail that applies to all Regions. To do so, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the name of the CloudWatch Logs log group that you associate with the CloudTrail trail. You create the metric filter for that log group.

Finally, create the metric filter and alarm.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Log groups**.
3. Select the check box for the CloudWatch Logs log group that is associated with the CloudTrail trail that you created.
4. From **Actions**, choose **Create Metric Filter**.
5. Under **Define pattern**, do the following:
 - a. Copy the following pattern and then paste it into the **Filter Pattern** field.

```
{($.eventName=CreateVpc) || ($.eventName>DeleteVpc) ||  
 ($.eventName=ModifyVpcAttribute) || ($.eventName=AcceptVpcPeeringConnection)  
 || ($.eventName>CreateVpcPeeringConnection)}
```

```
|| ($.eventName=DeleteVpcPeeringConnection) ||  
($.eventName=RejectVpcPeeringConnection) || ($.eventName=AttachClassicLinkVpc)  
|| ($.eventName=DetachClassicLinkVpc) || ($.eventName=DisableVpcClassicLink) ||  
($.eventName=EnableVpcClassicLink)}
```

- b. Choose **Next**.
6. Under **Assign metric**, do the following:
 - a. In **Filter name**, enter a name for your metric filter.
 - b. For **Metric namespace**, enter **LogMetrics**.

If you use the same namespace for all of your CIS log metric filters, then all CIS Benchmark metrics are grouped together.
 - c. For **Metric name**, enter a name for the metric. Remember the name of the metric. You will need to select the metric when you create the alarm.
 - d. For **Metric value**, enter **1**.
 - e. Choose **Next**.
7. Under **Review and create**, verify the information that you provided for the new metric filter. Then choose **Create metric filter**.
8. Choose the **Metric filters** tab, then choose the metric filter that you just created.

To choose the metric filter, select the check box at the upper right.
9. Choose **Create Alarm**.
10. Under **Specify metric and conditions**, do the following:
 - a. Under **Metric**, for **Statistic**, choose **Average**. For more information about the available statistics, see [Statistics](#) in the *Amazon CloudWatch User Guide*.
 - b. Under **Conditions**, for **Threshold**, choose **Static**.
 - c. For **Define the alarm condition**, choose **Greater/Equal**.
 - d. For **Define the threshold value**, enter **1**.
 - e. Choose **Next**.
11. Under **Configure actions**, do the following:
 - a. Under **Alarm state trigger**, choose **In alarm**.
 - b. Under **Select an SNS topic**, choose **Select an existing SNS topic**.
 - c. For **Send a notification to**, enter the name of the SNS topic that you created in the previous procedure.
 - d. Choose **Next**.
12. Under **Add name and description**, enter a **Name** and **Description** for the alarm. For example, **CIS-3.14-VPCCChanges**. Then choose **Next**.
13. Under **Preview and create**, review the alarm configuration. Then choose **Create alarm**.

4.1 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 22

Severity: High

AWS Config rule: [restricted-ssh](#)

Schedule type: Change triggered

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources.

CIS recommends that no security group allow unrestricted ingress access to port 22. Removing unfettered connectivity to remote console services, such as SSH, reduces a server's exposure to risk.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Remediation

Perform the following steps for each security group associated with a VPC.

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left pane, choose **Security groups**.
3. Select a security group.
4. In the bottom section of the page, choose the **Inbound Rules** tab.
5. Choose **Edit rules**.
6. Identify the rule that allows access through port 22 and then choose the **X** to remove it.
7. Choose **Save rules**.

4.2 – Ensure no security groups allow ingress from 0.0.0.0/0 to port 3389

Severity: High

AWS Config rule: `restricted-common-ports`

Schedule type: Change triggered

The name of the associated AWS Config managed rule is `restricted-common-ports`. However, the rule that is created uses the name `restricted-rdp`.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources.

CIS recommends that no security group allow unrestricted ingress access to port 3389. Removing unfettered connectivity to remote console services, such as RDP, reduces a server's exposure to risk.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Remediation

Perform the following steps for each security group associated with a VPC.

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the left pane, choose **Security groups**.
3. Select a security group.
4. In the bottom section of the page, choose the **Inbound Rules** tab.
5. Choose **Edit rules**.
6. Identify the rule that allows access through port 3389 and then choose the **X** to remove it.
7. Choose **Save rules**.

4.3 – Ensure the default security group of every VPC restricts all traffic

Severity: High

AWS Config rule: [vpc-default-security-group-closed](#)

Schedule type: Change triggered

A VPC comes with a default security group with initial settings that deny all inbound traffic, allow all outbound traffic, and allow all traffic between instances assigned to the security group. If you don't specify a security group when you launch an instance, the instance is automatically assigned to this default security group. Security groups provide stateful filtering of ingress and egress network traffic to AWS resources.

CIS recommends that the default security group restrict all traffic.

Update the default security group for the default VPC in every Region to comply. Any new VPCs automatically contain a default security group that you need to remediate to comply with this recommendation.

Note

When implementing this recommendation, you can use VPC flow logging, enabled for [the section called "2.9 – Ensure VPC flow logging is enabled in all VPCs "](#) (p. 366), to determine the least-privilege port access that systems require to work properly. VPC flow logging can log all packet acceptances and rejections that occur under the current security groups.

Configuring all VPC default security groups to restrict all traffic encourages least-privilege security group development and mindful placement of AWS resources into security groups. This in turn reduces the exposure of those resources.

Remediation

To update the default security group to restrict all access

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. View the default security groups details to see the resources that are assigned to them.
3. Create a set of least-privilege security groups for the resources.
4. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
5. On the Amazon EC2 console, change the security group for the resources that use the default security groups to the least-privilege security group you created.
6. For each default security group, choose the **Inbound** tab and delete all inbound rules.
7. For each default security group, choose the **Outbound** tab and delete all outbound rules.

For more information, see [Working with Security Groups](#) in the *Amazon VPC User Guide*.

CIS AWS Foundations Benchmark controls that you might want to disable

For the CIS AWS Foundations Benchmark standard, below are some specific controls that you might want to disable.

CIS AWS Foundations Benchmark 2.7 control

This control deals with using AWS KMS to encrypt CloudTrail trail logs. If you log these trails in a centralized logging account, you only need to run this control in the account and Region where centralized logging takes place.

CIS AWS Foundations Benchmark 1.2-1.14, 1.16, 1.20, 1.22, and 2.5 controls

To save on the cost of AWS Config, you can disable recording of global resources in all but one Region, and then disable these controls that deal with global resources in all Regions except for the Region that runs global recording.

If you disable these 1.x controls and disable recording of global resources in a particular Region, you should also disable 2.5. This is because 2.5 requires recording of global resources in order to pass.

CIS AWS Foundations Benchmark 1.1, 3.1-3.14

You might prefer to use Amazon GuardDuty for anomaly detection instead of CloudWatch alarms, which can be noisy.

If so, then you can disable these controls, which focus on CloudWatch alarms.

CIS AWS Foundations Benchmark security checks that are not supported in Security Hub

The following rules are *not* supported in the CIS AWS Foundations Benchmark standard in Security Hub, because they cannot be evaluated in an automated way. Security Hub focuses on automated security checks.

- 1.15 – Ensure security questions are registered in the AWS account
- 1.17 – Maintain current contact details
- 1.18 – Ensure security contact information is registered
- 1.19 – Ensure IAM instance roles are used for AWS resource access from instances
- 1.21 – Do not set up access keys during initial user setup for all IAM users that have a console password
- 4.4 – Ensure routing tables for VPC peering are "least access"

Payment Card Industry Data Security Standard (PCI DSS)

The Payment Card Industry Data Security Standard (PCI DSS) in Security Hub consists of a set of AWS security best practices controls. Each control applies to a specific AWS resource, and relates to one or more PCI DSS version 3.2.1 requirements. A PCI DSS requirement can be related to multiple controls. The details page for each PCI DSS control lists the specific PCI DSS requirements that are related to that control. See [the section called "Viewing details for a standard" \(p. 336\)](#).

The PCI DSS Compliance Standard in Security Hub is designed to help you with your ongoing PCI DSS security activities. The controls cannot verify whether your systems are compliant with the PCI DSS standard. They can neither replace internal efforts nor guarantee that you will pass a PCI DSS assessment. Security Hub does not check procedural controls that require manual evidence collection.

Security Hub currently scopes the controls at the account level. It is recommended that you enable these controls in all of your accounts that have resources that store, process, and/or transmit cardholder data.

This standard was validated by AWS Security Assurance Services LLC (AWS SAS), which is a team of Qualified Security Assessors (QSAs) certified to provide PCI DSS guidance and assessments by the PCI DSS Security Standards Council (PCI SSC). AWS SAS have confirmed that the automated checks can assist a customer in preparing for a PCI DSS assessment.

Contents

- [AWS Config resources required for PCI DSS controls \(p. 399\)](#)

- [PCI DSS controls \(p. 399\)](#)
- [PCI DSS controls that you might want to disable \(p. 456\)](#)

AWS Config resources required for PCI DSS controls

The PCI DSS controls perform checks against the following resources. For AWS Security Hub to accurately report findings for all of the controls, you must enable recording for these resources in AWS Config.

- AWS account
- `AWS::AutoScaling::AutoScalingGroup`
- `AWS::CloudTrail::Trail`
- `AWS::CodeBuild::Project`
- `AWS::DMS::ReplicationInstance`
- `AWS::EC2::EIP`
- `AWS::EC2::Instance`
- `AWS::EC2::SecurityGroup`
- `AWS::EC2::Volume`
- `AWS::EC2::VPC`
- `AWS::ElasticLoadBalancingV2::LoadBalancer`
- `AWS::Elasticsearch::Domain`
- `AWS::IAM::Policy`
- `AWS::IAM::User`
- `AWS::KMS::Key`
- `AWS::Lambda::Function`
- `AWS::RDS::DBInstance`
- `AWS::RDS::DBSnapshot`
- `AWS::Redshift::Cluster`
- `AWS::S3::Bucket`
- `AWS::SageMaker::NotebookInstance`
- `AWS::SSM::AssociationCompliance`
- `AWS::SSM::PatchCompliance`

PCI DSS controls

PCI DSS in Security Hub supports the following controls. For each control, the information includes the severity, the resource type, the AWS Config rule, and the remediation steps.

[\[PCI.AutoScaling.1\] Auto Scaling groups associated with a load balancer should use health checks](#)

Severity: Low

Resource type: `AWS::AutoScaling::AutoScalingGroup`

AWS Config rule: [autoscaling-group-elb-healthcheck-required](#)

Schedule type: Change triggered

Parameters: None

This control checks whether your Auto Scaling groups that are associated with a load balancer are using Elastic Load Balancing health checks.

PCI DSS does not require load balancing or highly available configurations. However, this check aligns with AWS best practices.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.2: Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.

Replicating systems using load balancing provides high availability and is a means to mitigate the effects of a DDoS event.

This is one method used to implement system hardening configurations.

Remediation

To enable Elastic Load Balancing health checks

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. To select the group from the list, choose the right box.
4. From **Actions**, choose **Edit**.
5. For **Health Check Type**, choose **ELB**.
6. For **Health Check Grace Period**, enter **300**.
7. Choose **Save**.

For more information on using a load balancer with an Auto Scaling group, see the [Amazon EC2 Auto Scaling User Guide](#).

[PCI.CloudTrail.1] CloudTrail logs should be encrypted at rest using AWS KMS keys

Severity: Medium

Resource type: AWS::CloudTrail::Trail

AWS Config rule: [cloud-trail-encryption-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether AWS CloudTrail is configured to use the server-side encryption (SSE) AWS KMS key encryption.

If you are only using the default encryption option, you can choose to disable this check.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 3.4: Render Primary Account Numbers (PAN) unreadable anywhere it is stored (including on portable digital media, backup media, and in logs).

If you are using AWS services to process and store PAN, your CloudTrail logs should be encrypted at rest. Encrypting logs ensures that if logs capture PAN(s), the PAN(s) are protected.

By default, the log files delivered by CloudTrail to your S3 bucket are encrypted using Amazon server-side encryption with Amazon S3-managed encryption keys (SSE-S3). See the [Amazon Simple Storage Service User Guide](#).

You can configure CloudTrail logs to leverage customer managed keys to further protect CloudTrail logs.

These are methods used to render PAN unreadable.

Remediation

To remediate this issue, you enable encryption for your CloudTrail log files.

For details on how to encrypt CloudTrail log files with AWS KMS managed keys (SSE-KMS), see [Encrypting CloudTrail log files with AWS KMS-managed keys \(SSE-KMS\)](#) in the *AWS CloudTrail User Guide*.

[PCI.CloudTrail.2] CloudTrail should be enabled

Severity: High

Resource type: AWS account

AWS Config rule: `cloudtrail-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail is enabled in your AWS account.

However, some AWS services do not enable logging of all APIs and events. You should implement any additional audit trails other than CloudTrail and review the documentation for each service in [CloudTrail Supported Services and Integrations](#).

Related PCI DSS requirements

This control is associated with the following PCI DSS requirements:

PCI DSS 10.1: Implement audit trails to link all access to system components to each individual user.

By enabling CloudTrail, Event History provides you with 90 days of readily available events and audit trails for access to system components by each individual user.

You can find the identity of the users in the `eventSource` section of the CloudTrail log.

PCI DSS 10.2.1: Implement automated audit trails for all system components to reconstruct the following events: All individual user accesses to cardholder data

Depending on where cardholder data is stored, individual user accesses to cardholder data could be found in the `userIdentity`, `eventSource`, `eventName`, or `responseElements` sections of the CloudTrail log.

PCI DSS 10.2.2: Implement automated audit trails for all system components to reconstruct the following events: All actions taken by any individual with root or administrative privileges

Root user identification is found in the `userIdentity` section of the log.

PCI DSS 10.2.3: Implement automated audit trails for all system components to reconstruct the following events: Access to all audit trails

Access to audit trails might be found in the `eventSource`, `eventName`, or `responseElements` sections of the log.

PCI DSS 10.2.4: Implement automated audit trails for all system components to reconstruct the following events: Invalid logical access attempts

You can find invalid logical access attempts in CloudTrail logs. For example: `responseElements` : "ConsoleLogin" and `responseElements` : "Failure".

PCI DSS 10.2.5: Implement automated audit trails for all system components to reconstruct the following events: Use of and changes to identification and authentication mechanisms—including but not limited to creation of new accounts and elevation of privileges—and all changes, additions, or deletions to accounts with root or administrative privileges

Use of and changes to identification and authentication mechanisms might be found in the `userAgent`, `eventName`, or `responseElements` sections of the log.

PCI DSS 10.2.6: Implement automated audit trails for all system components to reconstruct the following events: Initialization, stopping, or pausing of the audit logs

Starting and stopping logging is captured in the CloudTrail logs.

An example of audit log starting and stopping would look as follows within a CloudTrail Log:
`eventName` : "StopLogging" and `eventName` : "StartLogging"

PCI DSS 10.2.7: Implement automated audit trails for all system components to reconstruct the following events: Creation and deletion of system-level objects

Creation and deletion of system level-objects are captured in the CloudTrail logs. An example of a system-level object would be an AWS Lambda function.

CloudTrail captures the `createFunction` and `deleteFunction` API calls, as described in the [AWS Lambda Developer Guide](#).

PCI DSS 10.3.1: Record at least the following audit trail entries for all system components for each event: User identification

You can find user identification in the `userIdentity` section of the CloudTrail logs.

PCI DSS 10.3.2: Record at least the following audit trail entries for all system components for each event: Type of event

You can find the type of event in the `eventName` section of the CloudTrail log.

PCI DSS 10.3.3: Record at least the following audit trail entries for all system components for each event: Date and time

You can find the date and time of an event in the `eventTime` section of the CloudTrail log.

PCI DSS 10.3.4: Record at least the following audit trail entries for all system components for each event: Success or failure indication

You can find the success or failure indication in the `responseElements` section of the CloudTrail log.

PCI DSS 10.3.5: Record at least the following audit trail entries for all system components for each event: Origination of event

You can find the origination of an event in the `userAgent` or `sourceIPAddress` section of the CloudTrail log.

PCI DSS 10.3.6: Record at least the following audit trail entries for all system components for each event: Identity or name of affected data, system component, or resource.

You can find the identity of the resource in the `eventSource` section of the CloudTrail log.

Remediation

To create a new trail in CloudTrail

1. Sign in to the AWS Management Console using the IAM user you configured for CloudTrail administration.
2. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
3. In the **Region** selector, choose the AWS Region where you want your trail to be created. This is the Home Region for the trail.

The Home Region is the only AWS Region where you can view and update the trail after it is created, even if the trail logs events in all AWS Regions.

4. In the navigation pane, choose **Trails**.
5. On the **Trails** page, choose **Get Started Now**. If you do not see that option, choose **Create Trail**.
6. In **Trail name**, give your trail a name, such as **My-Management-Events-Trail**.

As a best practice, use a name that quickly identifies the purpose of the trail. In this case, you're creating a trail that logs management events.

7. In **Management Events**, make sure **Read/Write events** is set to **All**.
8. In **Data Events**, do not make any changes. This trail will not log any data events.
9. Create a new S3 bucket for the logs:
 - a. In **Storage Location**, in **Create a new S3 bucket**, choose **Yes**.
 - b. In **S3 bucket**, give your bucket a name, such as **my-bucket-for-storing-cloudtrail-logs**.

The name of your S3 bucket must be globally unique. For more information about S3 bucket naming requirements, see the [AWS CloudTrail User Guide](#).

- c. Under **Advanced**, choose **Yes** for both **Encrypt log files with SSE-KMS** and **Enable log file validation**.
10. Choose **Create**.

For more details, see the tutorial in the [AWS CloudTrail User Guide](#).

[PCI.CloudTrail.3] CloudTrail log file validation should be enabled

Severity: Low

Resource type: AWS::CloudTrail::Trail

AWS Config rule: `cloud-trail-log-file-validation-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail log file validation is enabled.

It does not check when configurations are altered.

To monitor and alert on log file changes, you can use Amazon EventBridge or CloudWatch metric filters.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 10.5.2: Protect audit trail files from unauthorized modifications.

CloudTrail log file validation creates a digitally signed digest file containing a hash of each log that CloudTrail writes to Amazon S3.

You can use these digest files to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log.

This is a method that helps to protect audit trail files from unauthorized modifications.

PCI DSS 10.5.5: Use file-integrity monitoring or change-detection software on logs to ensure that existing log data cannot be changed without generating alerts.

CloudTrail log file validation creates a digitally signed digest file containing a hash of each log that CloudTrail writes to Amazon S3.

You can use these digest files to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log.

This is a method that helps to ensure file-integrity monitoring or change-detection software is used on logs.

Remediation

To enable CloudTrail log file validation

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. In the **Name** column, choose the name of a trail to edit.
4. Under **General details**, choose **Edit**.
5. Under **Additional settings**, for **Log file validation**, select **Enabled**.
6. Choose **Save**.

[PCI.CloudTrail.4] CloudTrail trails should be integrated with CloudWatch Logs

Severity: Low

Resource type: AWS::CloudTrail::Trail

AWS Config rule: `cloud-trail-cloud-watch-logs-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail trails are configured to send logs to CloudWatch Logs.

It does not check for user permissions to alter logs or log groups. You should create specific CloudWatch rules to alert when CloudTrail logs are altered.

This control also does not check for any additional audit log sources other than CloudTrail being sent to a CloudWatch Logs group.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 10.5.3: Promptly back up audit trail files to a centralized log server or media that is difficult to alter.

CloudTrail uses Amazon S3 for log file storage and delivery, so log files are stored permanently.

CloudWatch Logs is a native way to promptly back up audit trail files.

Remediation

To ensure that CloudTrail trails are integrated with CloudWatch Logs

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Choose a trail that there is no value for in the **CloudWatch Logs Log group** column.
4. Scroll down to the **CloudWatch Logs** section and then choose **Edit**.
5. For **Log group** field, do one of the following:
 - To use the default log group, keep the name as is.
 - To use an existing log group, choose **Existing** and then enter the name of the log group to use.
 - To create a new log group, choose **New** and then enter a name for the log group to create.
6. Choose **Continue**.
7. For **IAM role**, do one of the following:
 - To use an existing role, choose **Existing** and then choose the role from the drop-down list.
 - To create a new role, choose **New** and then enter a name for the role to create.

The new role is assigned a policy that grants the necessary permissions.

To view the permissions granted to the role, expand the **Policy document**.
8. Choose **Save changes**.

For more information about configuring CloudWatch Logs monitoring with the console, see the [AWS CloudTrail User Guide](#).

[PCI.CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth

Severity: Critical

Resource type: AWS::CodeBuild::Project

AWS Config rule: `codebuild-project-source-repo-url-check`

Schedule type: Change triggered

Parameters: None

This control checks whether the GitHub or Bitbucket source repository URL contains either personal access tokens or a user name and password.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)

- Asia Pacific (Osaka)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 8.2.1: Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.

You can use CodeBuild in your PCI DSS environment to compile your source code, run unit tests, or produce artifacts that are ready to deploy. If you do, your authentication credentials should never be stored or transmitted in clear text or appear in the repository URL.

You should use OAuth instead of personal access tokens or a user name and password to grant authorization for accessing GitHub or Bitbucket repositories. This is a method to use strong cryptography to render authentication credentials unreadable.

Remediation

To remove basic authentication / (GitHub) Personal Access Token from CodeBuild Project Source

1. Open the CodeBuild console at <https://console.aws.amazon.com/codebuild/>.
2. Select your Build project that contains personal access tokens or a user name and password.
3. From **Edit**, choose **Source**.
4. Choose **Disconnect from GitHub / Bitbucket**.
5. Choose **Connect using OAuth** and then choose **Connect to GitHub / Bitbucket**.
6. In the message displayed by your source provider, authorize as appropriate.
7. Reconfigure your **Repository URL** and **additional configuration** settings, as needed.
8. Choose **Update source**.

To see CodeBuild use case-based samples, see the [AWS CodeBuild User Guide](#).

[PCI.CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials

Severity: Critical

Resource type: AWS::CodeBuild::Project

AWS Config rule: [codebuild-project-envvar-awscred-check](#)

Schedule: Change triggered

Parameters: None

This control checks whether the project contains environment variables `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 8.2.1: Using strong cryptography, render all authentication credentials (such as passwords/phrases) unreadable during transmission and storage on all system components.

You can use CodeBuild in your PCI DSS environment to compile your source code, runs unit tests, or produce artifacts that are ready to deploy. If you do, never store the authentication credentials `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` in clear text.

Using environmental variables to store credentials in your CodeBuild project may violate the requirement to use strong cryptography to render authentication credentials unreadable.

Remediation

To reference sensitive data in CodeBuild runtime using Environmental variables, use the following procedures.

To remove an Environmental Variable

1. Open the CodeBuild console at <https://console.aws.amazon.com/codebuild/>.
2. Expand **Build**, choose **Build project**, and then choose the build project that contains plaintext credentials.
3. From **Edit**, choose **Environment**.
4. Expand **Additional configuration** and then scroll to **Environment variables**.
5. Choose **Remove** next to the environment variable.
6. Choose **Update environment**.

To store sensitive values in the Amazon EC2 Systems Manager Parameter Store and then retrieve them from your build spec

1. Open the CodeBuild console at <https://console.aws.amazon.com/codebuild/>.
2. Expand **Build**, choose **Build project**, and then choose your build project that contains plaintext credentials.
3. From **Edit**, choose **Environment**.
4. Expand **Additional configuration** and then scroll to **Environment variables**.
5. In AWS Systems Manager, create a Systems Manager parameter that contains your sensitive data. For instructions on how to do this, refer to the tutorial in the [AWS Systems Manager User Guide](#).
6. After you create the parameter, copy the parameter name.
7. Back in the CodeBuild console, choose **Create environmental variable**.
8. For **name**, enter the name of your variable as it appears in your build spec.
9. For **value**, paste in the name of your parameter.
10. From **type**, choose **Parameter**.

11. Choose **Remove** next to your noncompliant environmental variable that contains plaintext credentials.
12. Choose **Update environment**.

See the information on environment variables in build environments in the [AWS CodeBuild User Guide](#).

[PCI.Config.1] AWS Config should be enabled

Severity: Medium

Resource type: AWS account

AWS Config rule: None. To run this check, Security Hub runs through audit steps prescribed for it in [Securing Amazon Web Services](#). No AWS Config managed rules are created in your AWS environment for this check.

Schedule type: Periodic

Parameters: None

This control checks whether AWS Config is enabled in the account for the local Region and is recording all resources.

It does not check for change detection for all critical system files and content files, as AWS Config supports only a subset of resource types.

The AWS Config service performs configuration management of supported AWS resources in your account and delivers log files to you. The recorded information includes the configuration item (AWS resource), relationships between configuration items, and any configuration changes between resources.

Security Hub recommends that you enable AWS Config in all Regions. The AWS configuration item history that AWS Config captures enables security analysis, resource change tracking, and compliance auditing.

Note

Because Security Hub is a Regional service, the check performed for this control checks only the current Region for the account. It does not check all Regions.

To allow security checks against global resources in each Region, you also must record global resources. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

For more information, see the [AWS Config Developer Guide](#).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 10.5.2: Protect audit trail files from unauthorized modifications.

AWS Config continuously monitors, tracks, and evaluates your [AWS resource configurations](#) for desired settings and generates configuration change history files every six hours.

You should enable AWS Config to protect audit trail files from unauthorized modifications.

PCI DSS 11.5: Deploy a change-detection mechanism to alert personnel to unauthorized modification of critical system files, configuration files, or content files; and configure the software to perform critical file comparisons at least weekly.

AWS Config continuously monitors, tracks, and evaluates your [AWS resource configurations](#) for desired settings and generates configuration change history files every six hours.

You should enable AWS Config to ensure a change-detection mechanism is deployed and is configured to perform critical file comparisons at least weekly.

Remediation

To configure AWS Config settings

1. Open the AWS Config console at <https://console.aws.amazon.com/config/>.
2. Select the Region to configure AWS Config in.
3. If you haven't used AWS Config before, see [Getting Started](#) in the *AWS Config Developer Guide*.
4. Navigate to the Settings page from the menu, and do the following:
 - Choose **Edit**.
 - Under **Resource types to record**, select **Record all resources supported in this region and Include global resources (e.g., AWS IAM resources)**.
 - Under **Data retention period**, choose the default retention period for AWS Config data, or specify a custom retention period.
 - Under **AWS Config role**, either choose **Create AWS Config service-linked role** or choose **Choose a role from your account** and then select the role to use.
 - Under **Amazon S3 bucket**, specify the bucket to use or create a bucket and optionally include a prefix.
 - Under **Amazon SNS topic**, select an Amazon SNS topic from your account or create one. For more information about Amazon SNS, see the [Amazon Simple Notification Service Getting Started Guide](#).
5. Choose **Save**.

For more information about using AWS Config from the AWS CLI, see the [AWS Config Developer Guide](#).

You can also use an AWS CloudFormation template to automate this process. For more information, see the [AWS CloudFormation User Guide](#).

[PCI.CW.1] A log metric filter and alarm should exist for usage of the "root" user

Severity: Critical

Resource type: AWS account

AWS Config rule: None. Security Hub runs through audit steps without creating an AWS Config managed rules in your AWS account for this check.

Schedule type: Periodic

Parameters: None

This control checks for the CloudWatch metric filters using the following pattern:

```
{ $.userIdentity.type = "Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent" }
```

It checks the following:

- The log group name is configured for use with active multi-Region CloudTrail.
- There is at least one Event Selector for a Trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`.
- There is at least one active subscriber to an Amazon SNS topic associated with the alarm.

Note

When Security Hub performs the check for this control, it looks for CloudTrail trails that the current account uses. These trails might be organization trails that belong to another account. Multi-Region trails also might be based in a different Region.

The check results in `FAILED` findings in the following cases:

- No trail is configured.
- The available trails that are in the current Region and that are owned by current account do not meet the control requirements.

The check results in a control status of `NO_DATA` in the following cases:

- The multi-Region trail is based in a different Region. Security Hub can only generate findings in the Region where the trail is based.
- The multi-Region trail belongs to a different account. Security Hub can only generate findings for the account that owns the trail.

For the alarm, the current account must either own the referenced Amazon SNS topic, or must get access to the Amazon SNS topic by calling `ListSubscriptionsByTopic`. Otherwise Security Hub generates `WARNING` findings for the control.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

The root user is the most privileged user in an AWS account and has unrestricted access to all resources in the AWS account.

You should set up log metric filters and alarms in the event that AWS account root user credentials are used.

You should also ensure that CloudTrail is enabled to keep an audit trail of actions taken by any individual with root or administrative privileges (see [the section called "\[PCI.CloudTrail.2\] CloudTrail should be enabled" \(p. 401\)](#)). Root user identification would be found in the `userIdentity` section of the CloudTrail log.

Remediation

The steps to remediate this issue include setting up an Amazon SNS topic, a metric filter, and an alarm for the metric filter.

These are the same steps to remediate findings for [the section called "3.3 – Ensure a log metric filter and alarm exist for usage of root user" \(p. 370\)](#).

To create an Amazon SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. Create an Amazon SNS topic that receives all CIS alarms.

Create at least one subscriber to the topic.

For more information about creating Amazon SNS topics, see the [Amazon Simple Notification Service Developer Guide](#).

3. Set up an active CloudTrail trail that applies to all Regions.

To do this, follow the remediation steps in [the section called “2.1 – Ensure CloudTrail is enabled in all Regions” \(p. 359\)](#).

Make a note of the associated log group name.

To create a metric filter and alarm

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. Choose **Logs**, then choose **Log groups**.
3. Choose the log group where CloudTrail is logging.
4. On the log group details page, choose **Metric filters**.
5. Choose **Create metric filter**.
6. Copy the following pattern and then paste it into **Filter pattern**.

```
{$.userIdentity.type="Root" && $.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent" }
```

7. Choose **Next**.
8. Enter the name of the new filter. For example, **RootAccountUsage**.
9. Confirm that the value for **Metric namespace** is **LogMetrics**.

This ensures that all CIS Benchmark metrics are grouped together.
10. In **Metric name**, enter the name of the metric.
11. In **Metric value**, enter **1**, and then choose **Next**.
12. Choose **Create metric filter**.
13. Next, set up the notification. Select the metric filter you just created, then choose **Create alarm**.
14. Enter the threshold for the alarm (for example, **1**), then choose **Next**.
15. Under **Select an SNS topic**, for **Send notification to**, choose an email list, then choose **Next**.
16. Enter a **Name** and **Description** for the alarm, such as **RootAccountUsageAlarm**, then choose **Next**.
17. Choose **Create Alarm**.

[PCI.DMS.1] AWS Database Migration Service replication instances should not be public

Severity: Critical

Resource type: AWS::DMS::ReplicationInstance

AWS Config rule: [dms-replication-not-public](#)

Schedule type: Periodic

Parameters: None

This control checks whether AWS DMS replication instances are public. To do this, it examines the value of the `PubliclyAccessible` field.

A private replication instance has a private IP address that you cannot access outside of the replication network. A replication instance should have a private IP address when the source and target databases are in the same network, and the network is connected to the replication instance's VPC using a VPN, AWS Direct Connect, or VPC peering. To learn more about public and private replication instances, see [Public and private replication instances](#) in the *AWS Database Migration Service User Guide*.

You should also ensure that access to your AWS DMS instance configuration is limited to only authorized users. To do this, restrict users' IAM permissions to modify AWS DMS settings and resources.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 1.2.1 - Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use AWS DMS in your defined CDE, set the replication instance's `PubliclyAccessible` field to `false`. Allowing public access to your replication instance might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1 - Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use AWS DMS in your defined CDE, set the replication instance's `PubliclyAccessible` field to `false`. Allowing public access to your replication instance might violate the requirement to limit inbound traffic to only system components that provide authorized, publicly accessible services, protocols, and ports.

PCI DSS 1.3.2 - Limit inbound internet traffic to IP addresses within the DMZ.

If you use AWS DMS in your defined CDE, set the replication instance's `PubliclyAccessible` field to `false`. Allowing public access to your replication instance might violate the requirement to limit inbound traffic to IP addresses within the DMZ.

PCI DSS 1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use AWS DMS in your defined CDE, set the replication instance's `PubliclyAccessible` field to `false`. Allowing public access to your replication instance might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use AWS DMS in your defined CDE, to migrate a database storing cardholder data, set the replication instance's `PubliclyAccessible` field to `false`. Allowing public access to your replication instance might violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

Remediation

Note that you cannot change the public access setting once a replication instance is created. It must be deleted and recreated.

To configure the AWS DMS replication instances setting to be not publicly accessible

1. Open the AWS Database Migration Service console at <https://console.aws.amazon.com/dms/>.
2. In the left navigation pane, under **Resource management**, navigate to **Replication instances**.

3. To delete the public instance, select the check box for the instance, choose **Actions**, then choose **delete**.
4. Choose **Create replication instance**. Provide the configuration details.
5. To disable public access, make sure that **Publicly accessible** is not selected.
6. Choose **Create**.

For more information, see the section on [Creating a replication instance](#) in the *AWS Database Migration Service User Guide*.

[PCI.EC2.1] Amazon EBS snapshots should not be publicly restorable

Severity: Critical

Resource type: AWS::EC2::Volume

AWS Config rule: [ebs-snapshot-publicly-restorable-check](#)

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic Block Store snapshots are not publicly restorable by everyone. Amazon EBS snapshots should not be publicly restorable by everyone unless you explicitly allow it, to avoid accidental exposure of your company's sensitive data.

You should also ensure that permission to change Amazon EBS configurations are restricted to authorized AWS accounts only. Learn more about managing Amazon EBS snapshot permissions in the [Amazon EC2 User Guide for Linux Instances](#).

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

Amazon EBS snapshots are used to back up the data on your Amazon EBS volumes to Amazon S3 at a specific point in time. They can be used to restore previous states of EBS volumes.

If an Amazon EBS snapshot stores cardholder data, it should not be publicly restorable by everyone. This would violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

Amazon EBS snapshots are used to back up the data on your Amazon EBS volumes to Amazon S3 at a specific point in time. They can be used to restore previous states of Amazon EBS volumes.

If an Amazon EBS snapshot stores cardholder data, it should not be publicly restorable by everyone. This would violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

Amazon EBS snapshots are used to back up the data on your Amazon EBS volumes to Amazon S3 at a specific point in time, and can be used to restore previous states of EBS volumes.

If an Amazon EBS snapshot stores cardholder data, it should not be publicly restorable by everyone. This would violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restrict access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

Amazon EBS snapshots are used to back up the data on your Amazon EBS volumes to Amazon S3 at a specific point in time. They can be used to restore previous states of Amazon EBS volumes.

If an Amazon EBS snapshot stores cardholder data, it should not be publicly restorable by everyone. This may violate the requirement to ensure access to systems components is restricted to least privilege necessary, or a user's need to know.

Remediation

To make a public Amazon EBS snapshot private

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Elastic Block Store**, choose **Snapshots** and then select your public snapshot.
3. Choose **Actions**, then choose **Modify permissions**
4. Choose **Private**
5. (Optional) Add AWS account numbers for authorized accounts to share your snapshot with.
6. Choose **Save**

For more information about sharing an Amazon EBS snapshot, see the [Amazon EC2 User Guide for Linux Instances](#).

[PCI.EC2.2] VPC default security group should prohibit inbound and outbound traffic

Severity: Medium

Resource type: AWS::EC2::SecurityGroup

AWS Config rule: `vpc-default-security-group-closed`

Schedule type: Change triggered

Parameters: None

This control checks that the default security group of a VPC does not allow inbound or outbound traffic.

It does not check for access restrictions for other security groups that are not default, and other VPC configurations.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If a service that is in scope for PCI DSS is associated with the default security group, the default rules for the security group will allow all outbound traffic. The rules also allow all inbound traffic from network interfaces (and their associated instances) that are assigned to the same security group.

You should change the default security group rules setting to restrict inbound and outbound traffic. Using the default might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If a service that is in scope for PCI DSS is associated with the default security group, the default rules for the security group will allow all outbound traffic. The rules also allow all inbound traffic from network interfaces (and their associated instances) that are assigned to the same security group.

You should change the default security group rules setting to restrict unauthorized inbound and outbound traffic. Using the default may violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 2.1: Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network.

If a service that is in scope for PCI DSS is associated with the default security group, the default rules for the security group will allow all outbound traffic. The rules also allow all inbound traffic from network interfaces (and their associated instances) that are assigned to the same security group.

You should change the default security group rules setting to restrict inbound and outbound traffic. Using the default may violate the requirement to remove or disable unnecessary default accounts.

Remediation

To remediate this issue, create new security groups and assign those security groups to your resources. To prevent the default security groups from being used, remove their inbound and outbound rules.

To create new security groups and assign them to your resources

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security groups**. View the default security groups details to see the resources that are assigned to them.
3. Create a set of least-privilege security groups for the resources. For details on how to create security groups, see [Creating a security group](#) in the *Amazon VPC User Guide*.
4. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
5. On the Amazon EC2 console, change the security group for the resources that use the default security groups to the least-privilege security group you created. See [Changing an instance's security groups](#) in the *Amazon VPC User Guide*.

After you assign the new security groups to the resources, remove the inbound and outbound rules from the default security groups. This ensures that the default security groups are not used.

To remove the rules from the default security group

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security groups**.
3. Select a default security group, and choose the **Inbound rules** tab. Choose **Edit inbound rules**. Then delete all of the inbound rules. Choose **Save rules**.

4. Repeat the previous step for each default security group.
5. Select a default security group and choose the **Outbound rules** tab. Choose **Edit outbound rules**. Then delete all of the outbound rules. Choose **Save rules**.
6. Repeat the previous step for each default security group.

For more information about working with security groups in Amazon VPC, see the [Amazon VPC User Guide](#).

[PCI.EC2.3] Unused EC2 security groups should be removed (Retired)

This control is retired.

[PCI.EC2.4] Unused EC2 EIPs should be removed

Severity: Low

Resource type: AWS : : EC2 : : EIP

AWS Config rule: [eip-attached](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Elastic IP addresses that are allocated to a VPC are attached to Amazon EC2 instances or in-use elastic network interfaces (ENIs).

A failed finding indicates you may have unused Amazon EC2 EIPs.

This will help you maintain an accurate asset inventory of EIPs in your cardholder data environment (CDE).

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.4: Maintain an inventory of system components that are in scope for PCI DSS.

If an EIP is not attached to an Amazon EC2 instance, this is an indication that it is no longer in use.

Unless there is a business need to retain them, you should remove unused resources to maintain an accurate inventory of system components.

Remediation

If you no longer need an Elastic IP address, Security Hub recommends that you release it (the address must not be associated with an instance).

To release an Elastic IP address using the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Network & Security**, choose **Elastic IPs**.
3. Choose the Elastic IP address, choose **Actions**, and then choose **Release Elastic IP address**.

4. When prompted, choose **Release**.

For more information, see the information on releasing Elastic IP addresses in the [Amazon EC2 User Guide for Linux Instances](#).

[PCI.EC2.5] Security groups should not allow ingress from 0.0.0.0/0 to port 22

Severity: High

Resource type: AWS::EC2::SecurityGroup

AWS Config rule: `restricted-ssh`

Schedule type: Change triggered

Parameters: None

This control checks whether security groups in use disallow unrestricted incoming SSH traffic.

It does not evaluate outbound traffic.

Note that security groups are stateful. If you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out regardless of outbound rules. To learn more about security groups, see [Security groups for your VPC](#) in the *Amazon VPC User Guide*.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1 - Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

You might allow SSH traffic to your instances that are in your defined CDE. If so, restrict the inbound SSH source from 0.0.0.0/0 (anywhere) to a specific IP address or range. Leaving unrestricted access to SSH might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1 - Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

You might allow SSH traffic to your instances that are in your defined CDE. If so, restrict the inbound SSH source from 0.0.0.0/0 (anywhere) to a specific IP address or range. Leaving unrestricted access to SSH might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 2.2.2 Enable only necessary services, protocols, daemons, etc., as required for the function of the system.

You might allow SSH traffic to your instances that are in your defined CDE. If so, restrict the inbound SSH source from 0.0.0.0/0 (anywhere) to a specific IP address or range as required for the function of the security group. Within a CDE, a security group could be considered a system component, which should be hardened appropriately. Leaving unrestricted access to SSH might violate the

requirement to enable only the necessary services, protocols, daemons, etc., that are required for the function of the system.

Remediation

Perform the following steps for each security group associated with a VPC.

To remove access to port 22 from a security group

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Security**, choose **Security groups**.
3. Select a security group.
4. In the bottom section of the page, choose **Inbound rules**.
5. Choose **Edit inbound rules**.
6. Identify the rule that allows access through port 22 and then choose the **X** to remove it.
7. Choose **Save rules**.

[PCI.EC2.6] VPC flow logging should be enabled in all VPCs

Severity: Medium

Resource type: AWS::EC2::VPC

AWS Config rule: [vpc-flow-logs-enabled](#)

Schedule type: Periodic

Parameters:

- `trafficType` – REJECT

This control checks whether VPC flow logs are found and enabled for VPCs. The traffic type is set to REJECT.

With VPC Flow Logs, you can capture information about the IP address traffic to and from network interfaces in your VPC. After you create a flow log, you can use CloudWatch Logs to view and retrieve the log data.

Security Hub recommends that you enable flow logging for packet rejects for VPCs. Flow logs provide visibility into network traffic that traverses the VPC. They can detect anomalous traffic and provide insight into security workflows.

By default, the record includes values for the different components of the IP address flow, including the source, destination, and protocol. For more information and descriptions of the log fields, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 10.3.3 Verify date and time stamp is included in log entries.

By enabling VPC flow logging for your VPC, you can identify the date and time of a log entry. The event date and time are recorded in the start and end fields. The values are displayed in Unix seconds.

PCI DSS 10.3.4 Verify success or failure indication is included in log entries.

By enabling VPC flow logging for your VPC, you can identify the type of event that occurred. The type of event is recorded in the action field, and can be either `ACCEPT` or `REJECT`.

PCI DSS 10.3.5 Verify origination of event is included in log entries.

By enabling VPC flow logging for your VPC, you can verify the origin of an event. The event origin is recorded in the `pkt-srcaddr`, `srcaddr`, and `srcport` fields. These fields show the source IP address and source port of the traffic.

PCI DSS 10.3.6 Verify identity or name of affected data, system component, or resources is included in log entries.

By enabling VPC flow logging for your VPC, you can verify the identity or name of affected data, system components, or resources. The `pkt-dstaddr`, `dstaddr`, and `dstport` fields show the destination IP address and destination port of the traffic.

Remediation

To remediate this issue, enable VPC flow logging.

To enable VPC flow logging

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Virtual Private Cloud**, choose **Your VPCs**.
3. Select a VPC to update.
4. At the bottom of the page, choose **Flow Logs**.
5. Choose **Create flow log**.
6. For **Filter**, choose **Reject**.
7. For **Destination log group**, choose the log group to use.
8. If you chose **CloudWatch Logs** for your destination log group, for **IAM role**, choose the IAM role to use.
9. Choose **Create**.

[PCI.ELBV2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

Severity: Medium

Resource type: `AWS::ElasticLoadBalancingV2::LoadBalancer`

AWS Config rule: `alb-http-to-https-redirection-check`

Schedule type: Periodic

Parameters: None

This control checks whether HTTP to HTTPS redirection is configured on all HTTP listeners of Application Load Balancers. The control fails if any of the HTTP listeners of Application Load Balancers do not have HTTP to HTTPS redirection configured.

Before you start to use your Application Load Balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners support both the HTTP and HTTPS protocols. You can use an HTTPS listener to offload the work of encryption and decryption to your load balancer. To enforce encryption in transit, you should use redirect actions with Application Load Balancers to redirect client HTTP requests to an HTTPS request on port 443.

To learn more, see [Listeners for your Application Load Balancers](#) in *User Guide for Application Load Balancers*.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.3 Encrypt all nonconsole administrative access using strong cryptography.

If you use Application Load Balancers with an HTTP listener, ensure that the listener is redirected to HTTPS for any nonconsole administrative access. Allowing unencrypted authentication over HTTP for administrators of the cardholder data environment might violate the requirement to encrypt all nonconsole administrative access using strong cryptography.

PCI DSS 4.1 Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks.

If you use Application Load Balancers with an HTTP listener, ensure that the listener is redirected to HTTPS for any transmissions of cardholder data. Allowing unencrypted transmissions of cardholder data might violate the requirement to use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks.

Remediation

To remediate this issue, you redirect HTTP request to HTTPS.

To redirect HTTP requests to HTTPS on an Application Load Balancer

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Load Balancing**, choose **Load balancers**.
3. Choose an Application Load Balancer.
4. Choose **Listeners**.
5. Select the check box for an HTTP listener (port 80 TCP) and then choose **Edit**.
6. If there is an existing rule, you must delete it. Otherwise, choose **Add action** and then choose **Redirect to...**
7. Choose **HTTPS** and then enter **443**.
8. Choose the check mark in a circle symbol and then choose **Update**.

[PCI.ES.1] Elasticsearch domains should be in a VPC

Severity: Critical

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: [elasticsearch-in-vpc-only](#)

Schedule type: Periodic

Parameters: None

This control checks whether Elasticsearch domains are in a VPC.

It does not evaluate the VPC subnet routing configuration to determine public reachability.

This AWS control also does not check whether the OpenSearch Service resource-based policy permits public access by other accounts or external entities. You should ensure that Elasticsearch domains are not attached to public subnets. See [Resource-based policies](#) in the *Amazon OpenSearch Service Developer Guide*.

You should also ensure that your VPC is configured according to the recommended best practices. See [Security best practices for your VPC](#) in the *Amazon VPC User Guide*.

Note

This control is not supported in Asia Pacific (Osaka).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If your OpenSearch Service clusters contain cardholder data, the OpenSearch Service domains should be placed in a VPC. Doing so enables secure communication between OpenSearch Service and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If your OpenSearch Service clusters contain cardholder data, the OpenSearch Service domains should be placed in a VPC. Doing so enables secure communication between OpenSearch Service and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If your OpenSearch Service clusters contain cardholder data, the OpenSearch Service domains should be placed in a VPC, which enables secure communication between OpenSearch Service and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to limit inbound internet traffic to IP addresses within the DMZ.

You can also use a resource-based policy and specify an IP condition for restricting access based on source IP addresses. See the blog post [How to control access to your Amazon OpenSearch Service domain](#).

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If your OpenSearch Service clusters contain cardholder data, the OpenSearch Service domains should be placed in a VPC, which enables secure communication between OpenSearch Service and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If your OpenSearch Service clusters contain cardholder data, the OpenSearch Service domains should be placed in a VPC. Doing so enables secure communication between OpenSearch Service and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

Remediation

If you create a domain with a public endpoint, you cannot later place it within a VPC. Instead, you must create a new domain and migrate your data.

The reverse is also true. If you create a domain within a VPC, it cannot have a public endpoint. Instead, you must either [create another domain](#) or disable this control.

See [Launching your Amazon OpenSearch Service domains within a VPC](#) in the *Amazon OpenSearch Service Developer Guide*.

[PCI.ES.2] Elasticsearch domains should have encryption at rest enabled

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: [elasticsearch-encrypted-at-rest](#)

Schedule type: Periodic

Parameters: None

This control checks whether Elasticsearch domains have encryption at rest configuration enabled.

Note

This control is not supported in Asia Pacific (Osaka).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 3.4: Render Primary Account Numbers (PAN) unreadable anywhere it is stored (including on portable digital media, backup media, and in logs).

If you use OpenSearch Service to store credit card Primary Account Numbers (PAN), the PAN should be protected by enabling OpenSearch Service domain encryption at rest.

If enabled, it encrypts the following aspects of a domain: Indices, automated snapshots, OpenSearch Service logs, swap files, all other data in the application directory.

This is a method used to render PAN unreadable.

Remediation

By default, domains do not encrypt data at rest, and you cannot configure existing domains to use the feature.

To enable the feature, you must create another domain and migrate your data. For information about creating domains, see the [Amazon OpenSearch Service Developer Guide](#).

Encryption of data at rest requires OpenSearch Service 5.1 or later. For more information about encrypting data at rest for OpenSearch Service, see the [Amazon OpenSearch Service Developer Guide](#).

[PCI.GuardDuty.1] GuardDuty should be enabled

Severity: High

Resource type: AWS account

AWS Config rule: `guardduty-enabled-centralized`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon GuardDuty is enabled in your AWS account and Region.

While GuardDuty can be effective against attacks that an intrusion detection system would typically protect, it might not be a complete solution for every environment. This rule also does not check for the generation of alerts to personnel. For more information about GuardDuty, see the [Amazon GuardDuty User Guide](#).

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- Middle East (Bahrain)
- AWS GovCloud (US-East)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 11.4 Use intrusion-detection and/or intrusion-prevention techniques to detect and/or prevent intrusions into the network.

GuardDuty can help to meet requirement 11.4 by monitoring traffic at the perimeter of the cardholder data environment and all critical points within it. It can also keep all intrusion-detection engines, baselines, and signatures up to date. Findings are generated from GuardDuty. You can send these alerts to personnel using Amazon CloudWatch. See [Creating custom responses to GuardDuty findings with Amazon CloudWatch Events](#) in the *Amazon GuardDuty User Guide*. Not enabling GuardDuty in your AWS account might violate the requirement to use intrusion-detection and/or prevention techniques to prevent intrusions into the network.

Remediation

To remediate this issue, you enable GuardDuty.

For details on how to enable GuardDuty, including how to use AWS Organizations to manage multiple accounts, see [Getting started with GuardDuty](#) in the *Amazon GuardDuty User Guide*.

[PCI.IAM.1] IAM root user access key should not exist

Severity: Critical

Resource type: AWS account

AWS Config rule: [iam-root-access-key-check](#)

Schedule type: Periodic

Parameters: None

This control checks whether user access keys exist for the root user.

Note

This control is not supported in Africa (Cape Town) or Asia Pacific (Osaka).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.1: Always change vendor-supplied defaults and remove or disable unnecessary default accounts before installing a system on the network.

The AWS account root user is the most privileged AWS user. AWS access keys provide programmatic access to a given account.

No access keys should be created for the root user, as this may violate the requirement to remove or disable unnecessary default accounts.

PCI DSS 2.2: Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.

The root user is the most privileged AWS user. AWS access keys provide programmatic access to a given account.

No access keys should be created for the root user, as this may violate the requirement to implement system hardening configurations.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

The root user is the most privileged AWS user. AWS access keys provide programmatic access to a given account.

No access keys should be created for the root user. Doing so might violate the requirement to ensure access to systems components is restricted to least privilege necessary, or a user's need to know.

Remediation

To delete access keys

1. Log in to your account using the root user credentials.
2. Choose the account name near the top-right corner of the page and then choose **My Security Credentials**.
3. In the pop-up warning, choose **Continue to Security Credentials**.
4. Choose **Access keys (access key ID and secret access key)**.
5. To permanently delete the key, choose **Delete** and then choose **Yes**. You cannot recover deleted keys.

6. If there is more than one root user access key, then repeat steps 4 and 5 for each key.

[PCI.IAM.2] IAM users should not have IAM policies attached

Severity: Low

Resource type: AWS : IAM : User

AWS Config rule: `iam-user-no-policies-check`

Schedule type: Change triggered

Parameters: None

This control checks that none of your IAM users have policies attached. IAM users must inherit permissions from IAM groups or roles.

It does not check whether least privileged policies are applied to IAM roles and groups.

Note

IAM users created by Amazon Simple Email Service are automatically created using inline policies. Security Hub automatically exempts these users from this control.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

IAM policies are how privileges are granted to users, groups, or roles in AWS.

By default, IAM users, groups, and roles have no access to AWS resources until IAM policies are attached to them.

To manage least privileged access and reduce the complexity of access management for PCI DSS in-scope resources, you should assign IAM policies at the group or role level and not at the user level.

Reducing access management complexity reduces opportunity for a principal to inadvertently receive or retain excessive privileges.

This is a method used to ensure access to systems components that contain cardholder data is restricted to least privilege necessary, or a user's need to know.

Remediation

To resolve this issue, do the following:

1. Create an IAM group
2. Assign the policy to the group
3. Add the users to the group

The policy is applied to each user in the group.

To create an IAM group

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.

2. Choose **Groups** and then choose **Create New Group**.
3. Enter a name for the group to create and then choose **Next Step**.
4. Select each policy to assign to the group and then choose **Next Step**.

The policies that you choose should include any policies currently attached directly to a user account. The next step to resolve a failed check is to add users to a group and then assign the policies to that group.

Each user in the group gets assigned the policies assigned to the group.

5. Confirm the details on the **Review** page and then choose **Create Group**.

For more information about creating IAM groups, see the [IAM User Guide](#).

To add users to an IAM group

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Groups**.
3. Choose **Group Actions** and then choose **Add Users to Group**.
4. Choose the users to add to the group and then choose **Add Users**.

For more information about adding users to groups, see the [IAM User Guide](#).

To remove a policy attached directly to a user

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. For the user to detach a policy from, in the **User name** column, choose the name.
4. For each policy listed under **Attached directly**, to remove the policy from the user, choose the **X** on the right side of the page and then choose **Remove**.
5. Confirm that the user can still use AWS services as expected.

[PCI.IAM.3] IAM policies should not allow full "*" administrative privileges

Severity: High

Resource type: AWS::IAM::Policy

AWS Config rule: [iam-policy-no-statements-with-admin-access](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the default version of AWS Identity and Access Management policies (also known as customer managed policies) do not have administrator access with a statement that has "Effect": "Allow" with "Action": "*" over "Resource": "*".

It only checks for the [customer managed policies](#) that you created, but does not check for full access to individual services, such as "S3:*".

It does not check for [inline and AWS managed policies](#).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restrict access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

Providing full administrative privileges instead of restricting to the minimum required may violate the requirement to ensure access to systems components is restricted to the least privilege necessary, or a user's need to know.

Remediation

To modify an IAM policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Policies**.
3. Choose the radio button next to the policy to remove.
4. From **Policy actions**, choose **Detach**.
5. On the **Detach policy** page, choose the radio button next to each user to detach the policy from and then choose **Detach policy**.
6. Confirm that the user that you detached the policy from can still access AWS services and resources as expected.

[PCI.IAM.4] Hardware MFA should be enabled for the root user

Severity: Critical

Resource type: AWS account

AWS Config rule: `root-account-hardware-mfa-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether your AWS account is enabled to use multi-factor authentication (MFA) hardware device to sign in with root user credentials.

It does not check whether you are using virtual MFA.

To address PCI DSS requirement 8.3.1, you can choose between hardware MFA (this control) or virtual MFA (the section called "[PCI.IAM.5] Virtual MFA should be enabled for the root user" (p. 428)).

Both time-based one-time password (TOTP) and Universal 2nd Factor (U2F) tokens are viable as hardware MFA options.

Note

This control is not supported in the following Regions.

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 8.3.1: Incorporate multi-factor authentication for all non-console access into the cardholder data environment (CDE) for personnel with administrative access.

The root user is the most privileged user in an account.

MFA adds an extra layer of protection on top of a user name and password. If users with administrative privileges are accessing the cardholder data environment over a network interface rather than via a direct, physical connection to the system component, and are not physically in front of the machine they are administering, MFA is required.

Enabling hardware MFA is a method used to incorporate multi-factor authentication (MFA) for all nonconsole administrative access

Remediation

To enable hardware-based MFA for the root user

1. Log in to your account using the root user credentials.
2. Choose the account name at the top right of the page and then choose **My Security Credentials**.
3. In the warning, choose **Continue to Security Credentials**.
4. Choose **Multi-factor authentication (MFA)**.
5. Choose **Activate MFA**.
6. Choose a hardware-based (not virtual) device to use for MFA and then choose **Continue**.
7. Complete the steps to configure the device type appropriate to your selection.

[PCI.IAM.5] Virtual MFA should be enabled for the root user

Severity: Critical

Resource type: AWS account

AWS Config rule: [root-account-mfa-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether users of your AWS account require a multi-factor authentication (MFA) device to sign in with root user credentials.

It does not check whether you are using hardware MFA.

To address PCI DSS requirement 8.3.1, you can choose between virtual MFA (this control) or hardware MFA (the section called “[PCI.IAM.4] Hardware MFA should be enabled for the root user” (p. 427)).

Note

This control is not supported in the following Regions.

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 8.3.1: Incorporate multi-factor authentication for all non-console access into the cardholder data environment (CDE) for personnel with administrative access.

The root user is the most privileged user in an account.

MFA adds an extra layer of protection on top of a user name and password. If users with administrative privileges are accessing the cardholder data environment, and are not physically in front of the machine they are administering, MFA is required.

Enabling virtual MFA is a method used to incorporate multi-factor authentication (MFA) for all nonconsole administrative access.

Remediation

To enable MFA for the root user

1. Log in to your account using the root user credentials.
2. Choose the account name at the top-right of the page and then choose **My Security Credentials**.
3. In the warning, choose **Continue to Security Credentials**.
4. Choose **Multi-factor authentication (MFA)**.
5. Choose **Activate MFA**.
6. Choose the type of device to use for MFA and then choose **Continue**.
7. Complete the steps to configure the device type appropriate to your selection.

[PCI.IAM.6] MFA should be enabled for all IAM users

Severity: Medium

Resource type: AWS::IAM::User

AWS Config rule: [iam-user-mfa-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether the IAM users have multi-factor authentication (MFA) enabled.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 8.3.1: Incorporate multi-factor authentication for all non-console access into the cardholder data environment (CDE) for personnel with administrative access.

Enabling MFA for all IAM users is a method used to incorporate multi-factor authentication (MFA) for all nonconsole administrative access.

Remediation

To configure MFA for a user

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. Choose the user name of the user to configure MFA for.

4. Choose **Security credentials** and then choose **Manage** next to **Assigned MFA device**.
5. Follow the **Manage MFA Device** wizard to assign the type of device appropriate for your environment.

To learn how to delegate MFA setup to users, the AWS Security Blog post [How to Delegate Management of Multi-Factor Authentication to AWS IAM Users](#).

[PCI.IAM.7] IAM user credentials should be disabled if not used within a predefined number of days

Severity: Medium

Resource type: AWS :: IAM :: User

AWS Config rule: [iam-user-unused-credentials-check](#)

Schedule type: Periodic

Parameters:

- `maxCredentialUsageAge: 90 (days)`

This control checks whether your IAM users have passwords or active access keys that have not been used within a specified number of days. The default is 90 days.

Security Hub strongly recommends that you do not generate and remove all access keys in your account. Instead, the recommended best practice is to either create one or more IAM roles or to use [federation](#). These practices allow your users to use their existing corporate credentials to sign in to the AWS Management Console console and AWS CLI.

Each approach has its use cases. Federation is generally better for enterprises that have an existing central directory or who plan to need more than the current quota of IAM users. Applications running outside of an AWS environment need access keys for programmatic access to AWS resources.

However, if the resources that need programmatic access run inside AWS, the best practice is to use IAM roles. You can use roles to grant a resource access without hardcoding an access key ID and secret access key into the configuration.

To learn more about protecting your access keys and account, see [Best practices for managing AWS access keys](#) in the *AWS General Reference*. Also see the blog post [Guidelines for protecting your AWS account while using-programmatic access](#).

If you already have an access key, we recommend that you remove or deactivate unused user credentials that are inactive for 90 days or longer.

This control only checks for inactive passwords or active access keys. It does not disable the account from use after 90 days. Customers are responsible for taking action and disabling the unused credentials.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 8.1.4 Remove/disable inactive user accounts within 90 days.

If you use IAM passwords or access keys, ensure that they are monitored for use, and disabled if not used for 90 days. Allowing IAM user accounts to remain active with unused credentials might violate the requirement to remove/disable inactive user accounts within 90 days.

Remediation

To get some of the information that you need to monitor accounts for dated credentials, use the IAM console. For example, when you view users in your account, there are columns for **Access key age**, **Password age**, and **Last activity**. If the value in any of these columns is greater than 90 days, make the credentials for those users inactive.

You can also use credential reports to monitor user accounts and identify those with no activity for 90 or more days. You can download credential reports in `.csv` format from the IAM console. For more information about credential reports, see [Getting credential reports for your AWS account](#) in the *IAM User Guide*.

After you identify the inactive accounts or unused credentials, use the following steps to disable them.

To disable inactive accounts or unused IAM credentials

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Under **Access management**, choose **Users**.
3. Choose the name of the user that has credentials older than 90 days.
4. Choose **Security credentials**. Choose **Make inactive** for all sign-in credentials and access keys that were not used in 90 days or more.

[PCI.IAM.8] Password policies for IAM users should have strong configurations

Severity: Medium

Resource type: AWS account

AWS Config rule: `iam-password-policy`

Schedule type: Periodic

Parameters: None

This control checks whether the account password policy for IAM users uses the following minimum PCI DSS configurations.

- `RequireUppercaseCharacters` – Require at least one uppercase character in password. (Default = `true`)
- `RequireLowercaseCharacters` – Require at least one lowercase character in password. (Default = `true`)
- `RequireNumbers` – Require at least one number in password. (Default = `true`)
- `MinimumPasswordLength` – Password minimum length. (Default = 7 or longer)
- `PasswordReusePrevention` – Number of passwords before allowing reuse. (Default = 4)
- `MaxPasswordAge` – Number of days before password expiration. (Default = 90)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 8.1.4: Remove/disable inactive user accounts within 90 days.

If you have IAM users in your AWS account, you should configure the IAM password policy appropriately. Not securing IAM users' passwords might violate the requirement to remove or disable inactive user accounts within 90 days. By default, the `MaxPasswordAge` parameter is set to 90 days.

After their password expires, IAM users cannot access their account until the password is changed, which disables the user.

PCI DSS 8.2.3: Passwords/passphrases must meet the following: Require a minimum length of at least seven characters and Contain both numeric and alphabetic characters.

If you have IAM users in your AWS account, the IAM password policy should be configured appropriately. Not securing IAM users' passwords might violate the requirement for a password to have a minimum length of at least seven characters. It might also violate the requirements to contain both numeric and alphabetic characters. By default, `MinimumPasswordLength` is 7, `RequireUppercaseCharacters` is true, and `RequireLowercaseCharacters` is true.

PCI DSS 8.2.4: Change user passwords/passphrases at least once every 90 days.

If you have IAM users in your AWS account, the IAM password policy should be configured appropriately. Not securing IAM users' passwords might violate the requirement to change user passwords or passphrases at least once every 90 days. By default, the `MaxPasswordAge` parameter is set to 90 days. After the password expires, the IAM user cannot access the account until the password is changed.

PCI DSS 8.2.5: Do not allow an individual to submit a new password/passphrase that is the same as any of the last four passwords/passphrases he or she has used.

If you have IAM users in your AWS account, the IAM password policy should be configured appropriately. Not securing IAM users' passwords might violate the requirement to not allow individuals to submit a new password or passphrase that is the same as any of their previous four passwords or passphrases. By default, `PasswordReusePrevention` is set to 4, which prevents users from reusing their last four passwords.

Remediation

You can use the IAM console to modify the password policy.

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Under **Access management**, choose **Account settings**.
3. Select **Prevent password reuse**. For **Number of passwords to remember**, enter **24**.
4. Choose **Change password policy**.
5. Select **Require at least one uppercase letter from Latin alphabet (A-Z)**.
6. Select **Require at least one lowercase letter from Latin alphabet (a-z)**.
7. Select **Require at least one non-alphanumeric character (!@#\$%^&*()_+~='[]|')**.
8. Select **Require at least one number**.
9. For **Enforce minimum password length**, enter **14**.
10. Select **Enable password expiration**. For **Expire passwords in day(s)**, enter **90**.
11. Choose **Save changes**.

[PCI.KMS.1] KMS key rotation should be enabled

Severity: Medium

Resource type: AWS::KMS::Key

AWS Config rule: [cmk-backing-key-rotation-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks that key rotation is enabled for each KMS key. It does not check KMS keys that have imported key material.

You should ensure keys that have imported material and those that are not stored in AWS KMS are rotated. AWS managed keys are rotated once every 3 years.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 3.6.4: Cryptographic keys should be changed once they have reached the end of their cryptoperiod.

While PCI DSS does not specify the time frame for cryptoperiods, if key rotation is enabled, rotation occurs annually by default.

If you use a KMS key to encrypt cardholder data, you should enable key rotation.

This is a method used to change cryptographic keys once they have reached the end of their cryptoperiod.

Remediation

To enable KMS key rotation

1. Open the AWS KMS console at <https://console.aws.amazon.com/kms>.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. Choose **Customer managed keys**.
4. In the **Alias** column, choose the alias of the key to update.
5. Choose **Key rotation**.
6. Select **Automatically rotate this KMS key every year** and then choose **Save**.

[PCI.Lambda.1] Lambda functions should prohibit public access

Severity: Critical

Resource type: `AWS::Lambda::Function`

AWS Config rule: `lambda-function-public-access-prohibited`

Schedule type: Change triggered

Parameters: None

This control checks whether the Lambda function resource-based policy prohibits public access.

It does not check for access to the Lambda function by internal principals, such as IAM roles. You should ensure that access to the Lambda function is restricted to authorized principals only by using least privilege Lambda resource-based policies.

For more information about using resource-based policies for AWS Lambda, see the [AWS Lambda Developer Guide](#).

Note

This control is not supported in the following Regions.

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use a Lambda function that is in scope for PCI DSS, the function should not be publicly accessible. A publicly accessible function might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use a Lambda function that is in scope for PCI DSS, the function should not be publicly accessible. A publicly accessible function might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If you use a Lambda function that is in scope for PCI DSS, the function should not be publicly accessible. A publicly accessible function might violate the requirement to limit inbound internet traffic to IP addresses within the DMZ.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use a Lambda function that is in scope for PCI DSS, the function must not be publicly accessible. A publicly accessible function might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

If you use a Lambda function that is in scope for PCI DSS, the function should not be publicly accessible. A publicly accessible function might violate the requirement to ensure access to systems components that contain cardholder data is restricted to the least privilege necessary, or a user's need to know.

Remediation

To remediate this issue, you update the resource-based policy to change the publicly accessible Lambda function to a private Lambda function.

You can only update resource-based policies for Lambda resources within the scope of the [AddPermission](#) and [AddLayerVersionPermission](#) API actions.

You cannot author policies for your Lambda resources in JSON, or use conditions that don't map to parameters for those actions using the CLI or the SDK.

To use the AWS CLI to revoke function-use permission from an AWS service or another account

1. To get the ID of the statement from the output of `GetPolicy`, from the AWS CLI, run the following:

```
aws lambda get-policy --function-name yourfunctionname
```

This command returns the Lambda resource-based policy string associated with the publicly accessible Lambda function.

2. From the policy statement returned by the `get-policy` command, copy the string value of the `Sid` field.
3. From the AWS CLI, run

```
aws lambda remove-permission --function-name yourfunctionname --statement-id youridvalue
```

To use the Lambda console to restrict access to the Lambda function

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. Navigate to **Functions** and then select your publicly accessible Lambda function.
3. Under **Designer**, choose the key icon at the top left. It has the tool-tip **View permissions**.
4. Under **Function policy**, if the policy allows actions for the principal element `"*" or {"AWS": "*"}`, it is publicly accessible.

Consider adding the following IAM condition to scope access to your account only.

```
"Condition": {  
  "StringEquals": {  
    "AWS:SourceAccount": "<account_id>"  
  }  
}
```

For other Lambda resource-based policies examples that allow you to grant usage permission to other accounts on a per-resource basis, see the information on using resource-based policies for AWS Lambda in the [AWS Lambda Developer Guide](#).

[PCI.Lambda.2] Lambda functions should be in a VPC

Severity: Low

Resource type: `AWS::Lambda::Function`

AWS Config rule: `lambda-inside-vpc`

Schedule type: Change triggered

Parameters: None

This control checks whether a Lambda function is in a VPC. You might see failed findings for `Lambda@Edge` resources.

It does not evaluate the VPC subnet routing configuration to determine public reachability.

Note

This control is not supported in the following Regions.

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

By default, Lambda runs your functions in a secure default VPC with access to AWS services and the internet.

If you use a Lambda function that is in scope for PCI DSS, the function can be configured to use a VPC endpoint. This would allow you to connect to your Lambda function from within a VPC without internet access. This method is used to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

By default, Lambda runs your functions in a secure default VPC with access to AWS services and the internet.

If you use a Lambda function that is in scope for PCI DSS, the function can be configured to use a VPC endpoint. This allows you to connect to your Lambda function from within a VPC without internet access. This is a method used to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

By default, Lambda runs your functions in a secure default VPC with access to AWS services and the internet.

If you use a Lambda function that is in scope for PCI DSS, the function can be configured to use a VPC endpoint. This allows you to connect to your Lambda function from within a VPC without internet access. This is a method used to limit inbound internet traffic to IP addresses within the DMZ.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

By default, Lambda runs your functions in a secure default VPC with access to AWS services and the internet.

If you use a Lambda function that is in scope for PCI DSS, the function can be configured to use a VPC endpoint. This allows you to connect to your Lambda function from within a VPC without internet access. This is a method used to block unauthorized outbound traffic from the cardholder data environment to the internet.

Remediation

To configure a function to connect to private subnets in a virtual private cloud (VPC) in your account

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. Navigate to **Functions** and then select your Lambda function.
3. Scroll to **Network** and then select a VPC with the connectivity requirements of the function.
4. To run your functions in high availability mode, Security Hub recommends that you choose at least two subnets.
5. Choose at least one security group that has the connectivity requirements of the function.
6. Choose **Save**.

For more information see the section on configuring a Lambda function to access resources in a VPC in the [AWS Lambda Developer Guide](#).

[PCI.OpenSearch.1] Amazon OpenSearch Service domains should be in a VPC

Category: Protect > Secure network configuration > Resources within VPC

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-in-vpc-only](#)

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains are in a VPC. It does not evaluate the VPC subnet routing configuration to determine public access.

You should ensure that OpenSearch domains are not attached to public subnets. See [Resource-based policies](#) in the Amazon OpenSearch Service Developer Guide.

You also should ensure that your VPC is configured according to the recommended best practices. See [Security best practices for your VPC](#) in the Amazon VPC User Guide.

OpenSearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls to secure access to OpenSearch domains, including network ACL and security groups. Security Hub recommends that you migrate public OpenSearch domains to VPCs to take advantage of these controls.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If your Amazon ES clusters contain cardholder data, the Amazon ES domains should be placed in a VPC. Doing so enables secure communication between Amazon ES and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If your Amazon ES clusters contain cardholder data, the Amazon ES domains should be placed in a VPC. Doing so enables secure communication between Amazon ES and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If your Amazon OpenSearch Service clusters contain cardholder data, the Amazon OpenSearch Service domains should be placed in a VPC, which enables secure communication between Amazon OpenSearch Service and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to limit inbound internet traffic to IP addresses within the DMZ.

You can also use a resource-based policy and specify an IP condition for restricting access based on source IP addresses. See the blog post [How to control access to your Amazon Elasticsearch Service domain](#).

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If your Amazon ES clusters contain cardholder data, the Amazon ES domains should be placed in a VPC, which enables secure communication between Amazon ES and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If your Amazon ES clusters contain cardholder data, the Amazon ES domains should be placed in a VPC. Doing so enables secure communication between Amazon ES and other services within the VPC without the need for an internet gateway, NAT device, or VPN connection port.

This method is used to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

Remediation

If you create a domain with a public endpoint, you cannot later place it within a VPC. Instead, you must create a new domain and migrate your data. The reverse is also true. If you create a domain within a VPC, it cannot have a public endpoint. Instead, you must either [create another domain](#) or disable this control.

See [Launching your Amazon OpenSearch Service domains within a VPC](#) in the Amazon OpenSearch Service Developer Guide.

[PCI.OpenSearch.2] OpenSearch domains should have encryption at rest enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: `opensearch-encrypted-at-rest`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon OpenSearch domains have encryption-at-rest configuration enabled. The check fails if encryption at rest is not enabled.

For an added layer of security for your sensitive data in OpenSearch, you should configure your OpenSearch domain to be encrypted at rest. OpenSearch domains offer encryption of data at rest. The feature uses AWS KMS to store and manage your encryption keys. To perform the encryption, it uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

To learn more about OpenSearch encryption at rest, see [Encryption of data at rest for Amazon OpenSearch Service](#) in the Amazon OpenSearch Service Developer Guide.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 3.4: Render Primary Account Numbers (PAN) unreadable anywhere it is stored (including on portable digital media, backup media, and in logs).

If you use Amazon OpenSearch Service to store credit card Primary Account Numbers (PAN), the PAN should be protected by enabling Amazon OpenSearch Service domain encryption at rest.

If enabled, it encrypts the following aspects of a domain: Indices, automated snapshots, Amazon OpenSearch Service logs, swap files, all other data in the application directory.

This is a method used to render PAN unreadable.

Remediation

By default, domains do not encrypt data at rest, and you cannot configure existing domains to use the feature. To enable the feature, you must create another domain and migrate your data.

For information about creating domains, see [Creating and managing Amazon OpenSearch Service domains](#) in the Amazon OpenSearch Service Developer Guide.

See [Launching your Amazon OpenSearch Service domains within a VPC](#) in the Amazon OpenSearch Service Developer Guide.

[PCI.RDS.1] Amazon RDS snapshots should prohibit public access

Severity: Critical

Resource type: AWS : :RDS : :DBSnapshot

AWS Config rule: `rds-snapshots-public-prohibited`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon RDS DB snapshots prohibit access by other accounts. You should also ensure that access to the snapshot and permission to change Amazon RDS configuration is restricted to authorized principals only.

To learn more about sharing DB snapshots in Amazon RDS, see the [Amazon RDS User Guide](#).

Note that if the configuration is changed to allow public access, the AWS Config rule may not be able to detect the change for up to 12 hours. Until the AWS Config rule detects the change, the check passes even though the configuration violates the rule.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

If an RDS snapshot stores cardholder data, the RDS snapshot should not be shared by other accounts. Sharing the RDS snapshot would allow other accounts to restore an RDS instance from the snapshot. Allowing this so might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

If an RDS snapshot stores cardholder data, the RDS snapshot should not be shared by other accounts. Sharing the RDS snapshot would allow other accounts to restore an RDS instance from the snapshot. Allowing this might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

If an RDS snapshot stores cardholder data, the RDS snapshot should not be shared by other accounts. Sharing the RDS snapshot would allow other accounts to restore an RDS instance from the snapshot. Allowing this might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

If an RDS snapshot stores cardholder data, the RDS snapshot should not be shared by other accounts. Sharing the RDS snapshot would allow other accounts to restore an RDS instance from the snapshot. Allowing this might violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

If an RDS snapshot stores cardholder data, the RDS snapshot should not be shared by other accounts. Sharing the RDS snapshot would allow other accounts to restore an RDS instance from the snapshot. Allowing this might violate the requirement to ensure access to systems components that contain cardholder data is restricted to least privilege necessary, or a user's need to know.

Remediation

To remove public access for Amazon RDS Snapshots

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. Navigate to **Snapshots** and then select the public Snapshot you want to modify
3. From the **Actions** list, choose **Share Snapshots**
4. From **DB snapshot visibility**, choose **Private**
5. Under **DB snapshot visibility**, select **for all**
6. Choose **Save**

[PCI.RDS.2] Amazon RDS DB Instances should prohibit public access, determined by PubliclyAccessible configuration

Severity: Critical

Resource type: AWS::RDS::DBInstance

AWS Config rule: `rds-instance-public-access-check`

Schedule type: Change triggered

Parameters: None

This control checks whether RDS instances are publicly accessible by evaluating the `publiclyAccessible` field in the instance configuration item. The value of `publiclyAccessible` indicates whether the DB instance is publicly accessible. When the DB instance is publicly accessible, it is an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. When the DB instance isn't publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address.

The control does not check VPC subnet routing settings or the Security Group rules. You should also ensure VPC subnet routing does not allow public access, and that the security group inbound rule associated with the RDS instance does not allow unrestricted access (0.0.0.0/0). You should also ensure that access to your RDS instance configuration is limited to only authorized users by restricting users' IAM permissions to modify RDS instances settings and resources.

For more information, see [Hiding a DB instance in a VPC from the Internet](#) in the *Amazon RDS User Guide*.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use an RDS instance that is in scope for PCI DSS, the RDS instance should not be publicly accessible. Allowing this might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use an RDS instance to store cardholder data, the RDS instance should not be publicly accessible. Allowing this might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If you use an RDS instance to store cardholder data, the RDS instance should not be publicly accessible as this might violate the requirement to limit inbound internet traffic to IP addresses within the DMZ.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use an RDS instance to store cardholder data, the RDS instance should not be publicly accessible. Allowing this might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use an RDS instance to store cardholder data, the RDS instance should not be publicly accessible. Allowing this may violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

If you use an RDS instance to store cardholder data, the RDS instance should not be publicly accessible, as this may violate the requirement to ensure access to systems components that contain cardholder data is restricted to least privilege necessary, or a user's need to know.

Remediation

To remove public access for Amazon RDS Databases

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Navigate to **Databases** and then choose your public database.
3. Choose **Modify**.
4. Scroll to **Network & Security**.
5. For **Public accessibility**, choose **No**.
6. Scroll to the bottom and then choose **Continue**.
7. Under **Scheduling of modifications**, choose **Apply immediately**.
8. Choose **Modify DB Instance**.

For more information about working with a DB Instance in a VPC, see the [Amazon RDS User Guide](#).

[PCI.Redshift.1] Amazon Redshift clusters should prohibit public access

Severity: Critical

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-cluster-public-access-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon Redshift clusters are publicly accessible by evaluating the `publiclyAccessible` field in the cluster configuration item.

Note

This control is not supported in Asia Pacific (Osaka).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use an Amazon Redshift cluster to store cardholder data, the cluster should not be publicly accessible. Allowing this might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use an Amazon Redshift cluster to store cardholder data, the cluster should not be publicly accessible. Allowing this might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If you use an Amazon Redshift cluster to store cardholder data, the cluster should not be publicly accessible, as this may violate the requirement to limit inbound internet traffic to IP addresses within the DMZ.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use an Amazon Redshift cluster to store cardholder data, the cluster should not be publicly accessible. Allowing this may violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use an Amazon Redshift cluster to store cardholder data, the cluster should not be publicly accessible. Allowing this might violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

Remediation

To disable public access for an Amazon Redshift cluster

1. Open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. On the navigation pane, choose **Clusters** and then select your public Amazon Redshift cluster.
3. From the **Cluster** drop-down menu, choose **Modify cluster**.
4. In **Publicly accessible**, choose **No**.
5. Choose **Modify**.

For more information about creating a cluster in a VPC, see the [Amazon Redshift Cluster Management Guide](#).

[PCI.S3.1] S3 buckets should prohibit public write access

Severity: Critical

Resource type: AWS :: S3 :: Bucket

AWS Config rule: [s3-bucket-public-write-prohibited](#)

Schedule type: Periodic and change triggered

Parameters: None

This control checks whether your S3 buckets allow public write access by evaluating the Block Public Access settings, the bucket policy, and the bucket access control list (ACL).

It does not check for write access to the bucket by internal principals, such as IAM roles. You should ensure that access to the bucket is restricted to authorized principals only.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public write access. Allowing public write access might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public write access. Allowing public write access might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

Unless you explicitly require everyone on the internet to be able to write to your S3 bucket, you should ensure that your S3 bucket is not publicly writable.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public write access. Allowing public write access might violate the requirement to limit inbound internet traffic to IP addresses within the DMZ.

PCI DSS 1.3.4: Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public write access. Allowing public write access might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public write access. Allowing public write access may violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public write access. Allowing public write access might violate the requirement to ensure access to systems components is restricted to least privilege necessary, or a user's need to know.

Remediation

To remove public access for an S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the name of the bucket identified in the finding.

3. Choose **Permissions** and then choose **Public access settings**.
4. Choose **Edit**, select all four options, and then choose **Save**.
5. If prompted, enter **confirm** and then choose **Confirm**.

[PCI.S3.2] S3 buckets should prohibit public read access

Severity: Critical

Resource type: AWS::S3::Bucket

AWS Config rule: `s3-bucket-public-read-prohibited`

Schedule type: Periodic and change triggered

Parameters: None

This control checks whether your S3 buckets allow public read access by evaluating the Block Public Access settings, the bucket policy, and the bucket access control list (ACL).

Unless you explicitly require everyone on the internet to be able to write to your S3 bucket, you should ensure that your S3 bucket is not publicly writable.

It does not check for read access to the bucket by internal principals, such as IAM roles. You should ensure that access to the bucket is restricted to authorized principals only.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1: Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public read access. Public read access might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1: Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public read access. Public read access might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2: Limit inbound internet traffic to IP addresses within the DMZ.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public read access. Public read access might violate the requirement to limit inbound internet traffic to IP addresses within the DMZ.

PCI DSS 1.3.6: Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public read access. Public read access might violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

PCI DSS 7.2.1: Establish an access control system(s) for systems components that restricts access based on a user's need to know, and is set to "deny all" unless specifically allowed. This access control system(s) must include the following: Coverage of all system components.

If you use an S3 bucket to store cardholder data, the bucket should prohibit public read access. Public read access might violate the requirement to ensure access to systems components is restricted to least privilege necessary, or a user's need to know.

Remediation

To remove public access for an S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the name of the bucket identified in the finding.
3. Choose **Permissions** and then choose **Public access settings**.
4. Choose **Edit**, select all four options, and then choose **Save**.
5. If prompted, enter **confirm** and then choose **Confirm**.

[PCI.S3.3] S3 buckets should have cross-region replication enabled

Severity: Low

Resource type: AWS::S3::Bucket

AWS Config rule: `s3-bucket-replication-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether S3 buckets have cross-region replication enabled.

PCI DSS does not require data replication or highly available configurations. However, this check aligns with AWS best practices for this control.

In addition to availability, you should consider other systems hardening settings.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.2: Develop configuration standards for all system components. Assure that these standards address all known security vulnerabilities and are consistent with industry-accepted system hardening standards.

Enabling cross-Region replication on S3 buckets ensures that multiple versions of the data are available in different distinct Regions. This allows you to store data at even greater distances, minimize latency, increase operational efficiency, and protect against DDoS and data corruption events.

This is one method used to implement system hardening configuration.

Remediation

To enable S3 bucket replication

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the S3 bucket that does not have cross-region replication enabled.
3. Choose **Management**, then choose **Replication**.
4. Choose **Add rule**. If versioning is not already enabled, you are prompted to enable it.
5. Choose your source bucket - **Entire bucket**.
6. Choose your destination bucket. If versioning is not already enabled on the destination bucket for your account, you are prompted to enable it.
7. Choose an IAM role. For more information on setting up permissions for replication, see the [Amazon Simple Storage Service User Guide](#).

8. Enter a rule name, choose **Enabled** for the status, then choose **Next**.
9. Choose **Save**.

For more information about replication, see the [Amazon Simple Storage Service User Guide](#).

[PCI.S3.4] S3 buckets should have server-side encryption enabled

Severity: Medium

Resource type: AWS : S3 : Bucket

AWS Config rule: `s3-bucket-server-side-encryption-enabled`

Schedule type: Change triggered

Parameters: None

This control checks that your Amazon S3 bucket either has Amazon S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server-side encryption.

When you set default encryption on a bucket, all new objects stored in the bucket are encrypted when they are stored, including clear text PAN data.

Server-side encryption for all of the objects stored in a bucket can also be enforced using a bucket policy. For more information about server-side encryption, see the [Amazon Simple Storage Service User Guide](#).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 3.4: Render Primary Account Numbers (PAN) unreadable anywhere it is stored (including on portable digital media, backup media, and in logs).

If you use an S3 bucket to store credit card Primary Account Numbers (PAN), then to render the PAN unreadable, the bucket default encryption should be enabled and/or the S3 bucket policy should explicitly deny put-object requests without server-side encryption.

Remediation

To enable default encryption on an S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the bucket from the list.
3. Choose **Properties**.
4. Under **Default encryption**, choose **Edit**.
5. Turn on Server-side encryption by choosing **Enable**. Then, choose either **SSE-S3** or **SSE-KMS**.
 - To use keys that are managed by Amazon S3 for default encryption, choose **SSE-S3**. For more information about using Amazon S3 server-side encryption to encrypt your data, see the [Amazon Simple Storage Service User Guide](#).
 - To use keys that are managed by AWS KMS for default encryption, choose **SSE-KMS**. Then choose a master key from the list of the AWS KMS master keys that you have created.

If you choose **SSE-KMS** type the Amazon Resource Name (ARN) of the AWS KMS key to use. You can find the ARN for your AWS KMS key in the IAM console, under **Encryption keys**. Or, you can choose a key name from the drop-down list.

Important

If you use the AWS KMS option for your default encryption configuration, you are subject to the RPS (requests per second) limits of AWS KMS. For more information about AWS KMS limits and how to request a limit increase, see the [AWS Key Management Service Developer Guide](#).

For more information about creating an AWS KMS key, see the [AWS Key Management Service Developer Guide](#).

For more information about using AWS KMS with Amazon S3, see the [Amazon Simple Storage Service User Guide](#).

When enabling default encryption, you might need to update your bucket policy. For more information about moving from bucket policies to default encryption, see the [Amazon Simple Storage Service User Guide](#).

6. Choose **Save**.

For more information about default S3 bucket encryption, see the [Amazon Simple Storage Service User Guide](#).

[PCI.S3.5] S3 buckets should require requests to use Secure Socket Layer

Severity: Medium

Resource type: AWS::S3::Bucket

AWS Config rule: `s3-bucket-ssl-requests-only`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon S3 buckets have policies that require requests to use Secure Socket Layer (SSL).

S3 buckets should have policies that require all requests (`Action: S3:*`) to only accept transmission of data over HTTPS in the S3 resource policy, indicated by the condition key `aws:SecureTransport`.

This does not check the SSL or TLS version. You should not allow early versions of SSL or TLS (SSLv3, TLS1.0) per PCI DSS requirements.

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 4.1 Use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks.

If you use S3 buckets to store cardholder data, ensure that bucket policies require that requests to the bucket only accept transmission of data over HTTPS. For example, you could use the policy statement `"aws:SecureTransport": "false"` to deny any requests not accessed through HTTPS. Allowing unencrypted transmissions of cardholder data might violate the requirement to use strong cryptography and security protocols to safeguard sensitive cardholder data during transmission over open, public networks.

Remediation

To remediate this issue, update the permissions policy of the S3 bucket.

To configure an S3 bucket to deny nonsecure transport

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Navigate to the noncompliant bucket, and then choose the bucket name.
3. Choose **Permissions**, then choose **Bucket Policy**.
4. Add a similar policy statement to that in the policy below. Replace `awsexamplebucket` with the name of the bucket you are modifying.

```
{
  "Id": "ExamplePolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSSLRequestsOnly",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::awsexamplebucket",
        "arn:aws:s3:::awsexamplebucket/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      },
      "Principal": "*"
    }
  ]
}
```

5. Choose **Save**.

For more information, see the knowledge center article [What S3 bucket policy should I use to comply with the AWS Config rule s3-bucket-ssl-requests-only?](#)

[PCI.S3.6] S3 Block Public Access setting should be enabled

Severity: Medium

Resource type: S3 AWS account

AWS Config rule: `s3-account-level-public-access-blocks-periodic`

Schedule type: Periodic

Parameters:

- `ignorePublicAcls: true`
- `blockPublicPolicy: true`
- `blockPublicAcls: true`
- `restrictPublicBuckets: true`

This control checks whether the following public access block settings are configured at the account level.

- `ignorePublicAcls: true,`
- `blockPublicPolicy: true`
- `blockPublicAcls: true`

- `restrictPublicBuckets: true`

The control passes if all of the public access block settings are set to `true`.

The control fails if any of the settings are set to `false`, or if any of the settings are not configured.

As an AWS best practice, S3 buckets should block public access. Unless you explicitly require everyone on the internet to be able to access your S3 bucket, you should ensure that your S3 bucket is not publicly accessible.

Note

This control is not supported in the following Regions.

- Asia Pacific (Osaka)
- Europe (Milan)
- Middle East (Bahrain)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 1.2.1 - Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use S3 buckets to store cardholder data, ensure that the bucket does not allow public access. Public access to your S3 bucket might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1 - Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use S3 buckets to store cardholder data, ensure that the bucket does not allow public access. Allowing public access to your S3 bucket might violate the requirement to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2 - Limit inbound internet traffic to IP addresses within the DMZ.

If you use S3 buckets to store cardholder data, ensure that the bucket does not allow public access. Allowing public access to your S3 bucket might violate the requirement to limit inbound traffic to IP addresses within the DMZ.

PCI DSS 1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use S3 buckets to store cardholder data, ensure that the bucket does not allow public access. Allowing public access to your S3 bucket might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use S3 buckets to store cardholder data, ensure that the bucket does not allow public access. Allowing public access to your S3 bucket might violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

Remediation

To enable Amazon S3 Block Public Access

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.

2. In the navigation pane, choose **Block public access (account settings)**.
3. Choose **Edit**. Then select **Block all public access**.
4. Choose **Save changes**.

For more information, see [Using Amazon S3 block public access](#) in the *Amazon Simple Storage Service User Guide*.

[PCI.SageMaker.1] Amazon SageMaker notebook instances should not have direct internet access

Severity: High

Resource type: AWS::SageMaker::NotebookInstance

AWS Config rule: [sagemaker-notebook-no-direct-internet-access](#)

Schedule type: Periodic

Parameters: None

This control checks whether direct internet access is disabled for an SageMaker notebook instance. To do this, it checks whether the `DirectInternetAccess` field is disabled for the notebook instance.

If you configure your SageMaker instance without a VPC, then by default direct internet access is enabled on your instance. You should configure your instance with a VPC and change the default setting to **Disable — Access the internet through a VPC**.

To train or host models from a notebook, you need internet access. To enable internet access, make sure that your VPC has a NAT gateway and your security group allows outbound connections. To learn more about how to connect a notebook instance to resources in a VPC, see [Connect a notebook instance to resources in a VPC](#) in the *Amazon SageMaker Developer Guide*.

You should also ensure that access to your SageMaker configuration is limited to only authorized users. Restrict users' IAM permissions to modify SageMaker settings and resources.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- AWS GovCloud (US-East)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 1.2.1 - Restrict inbound and outbound traffic to that which is necessary for the cardholder data environment (CDE), and specifically deny all other traffic.

If you use SageMaker notebook instances within your CDE, ensure that the notebook instance does not allow direct internet access. Allowing direct public access to your notebook instance might violate the requirement to allow only necessary traffic to and from the CDE.

PCI DSS 1.3.1 - Implement a DMZ to limit inbound traffic to only system components that provide authorized publicly accessible services, protocols, and ports.

If you use SageMaker notebook instances within your CDE, ensure that the notebook instance does not allow direct internet access. Allowing direct public access to your notebook instance might violate the requirement to only allow access to system components that provide authorized publicly accessible services, protocols, and ports.

PCI DSS 1.3.2 - Limit inbound internet traffic to IP addresses within the DMZ.

If you use SageMaker notebook instances within your CDE, ensure that the notebook instance does not allow direct internet access. Allowing direct public access to your notebook instance might violate the requirement to limit inbound traffic to IP addresses within the DMZ.

PCI DSS 1.3.4 Do not allow unauthorized outbound traffic from the cardholder data environment to the internet.

If you use SageMaker notebook instances within your CDE, ensure that the notebook instance does not allow direct internet access. Allowing direct public access to your notebook instance might violate the requirement to block unauthorized outbound traffic from the cardholder data environment to the internet.

PCI DSS 1.3.6 Place system components that store cardholder data (such as a database) in an internal network zone, segregated from the DMZ and other untrusted networks.

If you use SageMaker notebook instances, and the notebook instance contains cardholder data, restrict direct internet access. Allowing direct public access to your notebook instance might violate the requirement to place system components that store cardholder data in an internal network zone, segregated from the DMZ and other untrusted networks.

Remediation

Note that you cannot change the internet access setting after a notebook instance is created. It must be stopped, deleted, and recreated.

To configure an SageMaker notebook instance to deny direct internet access

1. Open the SageMaker console at <https://console.aws.amazon.com/sagemaker/>
2. Navigate to **Notebook instances**.
3. Delete the instance that has direct internet access enabled. Choose the instance, choose **Actions**, then choose **stop**.

After the instance is stopped, choose **Actions**, then choose **delete**.

4. Choose **Create notebook instance**. Provide the configuration details.
5. Expand the **Network** section. Then choose a VPC, subnet, and security group. Under **Direct internet access**, choose **Disable — Access the internet through a VPC**.
6. Choose **Create notebook instance**.

For more information, see [Connect a notebook instance to resources in a VPC](#) in the *Amazon SageMaker Developer Guide*.

[PCI.SSM.1] Amazon EC2 instances managed by Systems Manager should have a patch compliance status of COMPLIANT after a patch installation

Severity: Medium

Resource type: AWS::SSM::PatchCompliance and AWS::EC2::Instance

AWS Config rule: `ec2-managedinstance-patch-compliance-status-check`

Schedule: Change triggered

Parameters: None

This control checks whether the compliance status of the Amazon EC2 Systems Manager patch compliance is `COMPLIANT` or `NON_COMPLIANT` after the patch installation on the instance.

It only checks instances that are managed by AWS Systems Manager Patch Manager.

It does not check whether the patch was applied within the 30-day limit prescribed by PCI DSS requirement 6.2.

It also does not validate whether the patches applied were classified as security patches.

You should create patching groups with the appropriate baseline settings and ensure in-scope systems are managed by those patch groups in Systems Manager. For more information about patch groups, see the *AWS Systems Manager User Guide*.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 6.2: Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor-supplied security patches. Install critical security patches within one month of release.

Patches released by the vendor for systems that are in-scope for PCI DSS should be tested and validated before installation in production environment. Once deployed, security settings and controls should be validated to ensure that deployed patches have not impacted the security of the cardholder data environment (CDE).

If you use Amazon EC2 instances managed by AWS Systems Manager Patch Manager to patch managed instances in your CDE, ensure that the patches are successfully applied. To do this, check that the compliance status of the Amazon EC2 Systems Manager patch compliance is "COMPLIANT". Patch Manager can apply both operating systems and applications applicable patches.

This is a method used to protect system components and software from known vulnerabilities.

Remediation

To remediate noncompliant patches

This rule checks whether the compliance status of the Amazon EC2 Systems Manager patch compliance is `COMPLIANT` or `NON_COMPLIANT`. To find out more about patch compliance states, see the *AWS Systems Manager User Guide*.

1. Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>.
2. In the navigation pane, under **Node Management**, choose **Run Command**.

3. Choose **Run command**.
4. Choose the radio button next to `AWS-RunPatchBaseline` and then change the **Operation** to **Install**.
5. Choose **Choose instances manually** and then choose the noncompliant instance(s).
6. Scroll to the bottom and then choose **Run**.
7. After the command has completed, to monitor the new compliance status of your patched instances, in the navigation pane, choose **Compliance**.

See the *AWS Systems Manager User Guide* for more information about the following

- [Using Systems Manager documents to patch a managed instance](#)
- [Running commands using the Systems Manager Run command](#)

[PCI.SSM.2] Instances managed by Systems Manager should have an association compliance status of COMPLIANT

Severity: Low

Resource type: `AWS::SSM::AssociationCompliance`

AWS Config rule: `ec2-managedinstance-association-compliance-status-check`

Schedule type: Change triggered

Parameters: None

This control checks whether the status of the AWS Systems Manager association compliance is `COMPLIANT` or `NON_COMPLIANT` after the association is run on an instance. The control passes if the association compliance status is `COMPLIANT`.

A State Manager association is a configuration that is assigned to your managed instances. The configuration defines the state that you want to maintain on your instances. For example, an association can specify that antivirus software must be installed and running on your instances, or that certain ports must be closed.

After you create one or more State Manager associations, compliance status information is immediately available to you in the console or in response to AWS CLI commands or corresponding Systems Manager API operations. For associations, **Configuration Compliance** shows statuses of **Compliant** or **Non-compliant** and the severity level assigned to the association, such as **Critical** or **Medium**. To learn more about State Manager association compliance, see [About State Manager association compliance](#) in the *AWS Systems Manager User Guide*.

You must configure your in-scope EC2 instances for Systems Manager association. You must also configure the patch baseline for the security rating of the vendor of patches, and set the autoapproval date to meet PCI DSS 3.2.1 requirement 6.2. For additional guidance on how to create an association, see [Create an association](#) in the *AWS Systems Manager User Guide*. For additional information on working with patching in Systems Manager, see [AWS Systems Manager Patch Manager](#) in the *AWS Systems Manager User Guide*.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)

Related PCI DSS requirements

This control is related to the following PCI DSS requirements.

PCI DSS 2.4 Maintain an inventory of system components that are in scope for PCI DSS.

If you use EC2 instances managed by Systems Manager to collect inventory for your cardholder data environment (CDE), make sure that the instances are successfully associated. To do this, check whether the compliance status of the Systems Manager association compliance is `COMPLIANT`. Using Systems Manager can help to maintain an inventory of system components that are in scope for PCI DSS. For additional guidance on how to organize inventory, see [Configuring Resource Data Sync for Inventory](#) in the *AWS Systems Manager User Guide*.

Remediation

A failed association can be related to different things, including targets and SSM document names. To remediate this issue, you must first identify and investigate the association. You can then update the association to correct the specific issue.

You can edit an association to specify a new name, schedule, severity level, or targets. After you edit an association, Systems Manager creates a new version.

To investigate and update a failed association

1. Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>.
2. In the navigation pane, under **Node Management**, choose **Fleet Manager**.
3. Choose the instance ID that has an **Association status** of **Failed**.
4. Choose **View details**.
5. Choose **Associations**.
6. Note the name of the association that has an **Association status** of **Failed**. This is the association that you need to investigate. You need to use the association name in the next step.
7. In the navigation pane, under **Node Management**, choose **State Manager**. Search for the association name, then select the association.

After you determine the issue, edit the failed association to correct the problem. For information on how to edit an association, see [Edit an association](#).

For more information on creating and editing State Manager associations, see [Working with associations in Systems Manager](#) in the *AWS Systems Manager User Guide*.

[PCI.SSM.3] EC2 instances should be managed by AWS Systems Manager

Severity: Medium

Resource type: `AWS::EC2::Instance`

AWS Config rule: [ec2-instance-managed-by-systems-manager](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the EC2 instances in your account are managed by Systems Manager.

AWS Systems Manager is an AWS service that you can use to view and control your AWS infrastructure. To help you to maintain security and compliance, Systems Manager scans your managed instances. A managed instance is a machine that is configured for use with Systems Manager. Systems Manager then

reports or takes corrective action on any policy violations that it detects. Systems Manager also helps you to configure and maintain your managed instances. Additional configuration is needed in Systems Manager for patch deployment to managed EC2 instances.

To learn more, see the [AWS Systems Manager User Guide](#).

Related PCI DSS requirements

This control is related to the following PCI DSS requirements:

PCI DSS 2.4 Maintain an inventory of system components that are in scope for PCI DSS.

If you use EC2 instances managed by Systems Manager to collect inventory for your cardholder data environment (CDE), make sure that the instances are managed by Systems Manager. Using Systems Manager can help to maintain an inventory of system components that are in scope for PCI DSS. For additional guidance on how to organize inventory, see [Configuring resource data sync for inventory](#) in the *AWS Systems Manager User Guide*.

PCI DSS 6.2 Ensure that all system components and software are protected from known vulnerabilities by installing applicable vendor supplied security patches. Install critical security patches within one month of release.

For systems that are in scope for PCI DSS, before you install vendor patches in a production environment, you should test and validate them. After you deploy patches, validate security settings and controls to ensure that deployed patches have not affected the security of the CDE. If you use EC2 instances managed by Systems Manager to patch managed instances in your CDE, ensure that the instances are managed by Systems Manager. Systems Manager deploys system patches, which helps to protect system components and software from known vulnerabilities.

Remediation

You can use the Systems Manager quick setup to set up Systems Manager to manage your EC2 instances.

To determine whether your instances can support Systems Manager associations, see [Systems Manager prerequisites](#) in the *AWS Systems Manager User Guide*.

To ensure EC2 instances are managed by Systems Manager

1. Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>.
2. In the navigation pane, choose **Quick setup**.
3. On the configuration screen, keep the default options.
4. Choose **Set up Systems Manager**.

PCI DSS controls that you might want to disable

To save on the cost of AWS Config, you can disable recording of global resources in all but one Region. Then disable these controls that deal with global resources in all Regions except for the Region that runs global recording.

- the section called “[PCI.CW.1] A log metric filter and alarm should exist for usage of the “root” user” (p. 409)
- the section called “[PCI.IAM.1] IAM root user access key should not exist” (p. 424)
- the section called “[PCI.IAM.2] IAM users should not have IAM policies attached” (p. 425)
- the section called “[PCI.IAM.3] IAM policies should not allow full “*” administrative privileges” (p. 426)
- the section called “[PCI.IAM.4] Hardware MFA should be enabled for the root user” (p. 427)
- the section called “[PCI.IAM.5] Virtual MFA should be enabled for the root user” (p. 428)

- [the section called “\[PCI.IAM.6\] MFA should be enabled for all IAM users” \(p. 429\)](#)

If you disable these controls and disable recording of global resources in a particular Region, you should also disable [\[PCI.Config.1\] AWS Config should be enabled \(p. 408\)](#). This is because [\[PCI.Config.1\] AWS Config should be enabled \(p. 408\)](#) requires recording of global resources in order to pass.

AWS Foundational Security Best Practices standard

The AWS Foundational Security Best Practices standard is a set of controls that detect when your deployed accounts and resources deviate from security best practices.

The standard allows you to continuously evaluate all of your AWS accounts and workloads to quickly identify areas of deviation from best practices. It provides actionable and prescriptive guidance on how to improve and maintain your organization’s security posture.

The controls include best practices from across multiple AWS services. Each control is assigned a category that reflects the security function that it applies to. See [the section called “Control categories” \(p. 599\)](#).

Topics

- [AWS Config resources required for AWS Foundational Security Best Practices controls \(p. 457\)](#)
- [AWS Foundational Security Best Practices controls \(p. 459\)](#)
- [Control categories \(p. 599\)](#)
- [AWS Foundational Best Practices controls that you might want to disable \(p. 601\)](#)

AWS Config resources required for AWS Foundational Security Best Practices controls

AWS Foundational Security Best Practices controls involve queries against the following resources. For AWS Security Hub to accurately report findings for all of the controls, you must enable recording for these resources in AWS Config.

Note

In Regions where a control is not available, the corresponding resource is not available in AWS Config.

- AWS account
- `AWS::ACM::Certificate`
- `AWS::ApiGateway::Stage`
- `AWS::ApiGatewayV2::Stage`
- `AWS::AutoScaling::AutoScalingGroup`
- `AWS::AutoScaling::LaunchConfiguration`
- `AWS::CloudFormation::Stack`
- `AWS::CloudFront::Distribution`
- `AWS::CloudTrail::Trail`
- `AWS::CodeBuild::Project`
- `AWS::DAX::Cluster`
- `AWS::DMS::ReplicationInstance`
- `AWS::DynamoDB::Table`
- `AWS::EC2::Instance`
- `AWS::EC2::NetworkAcl`

- AWS::EC2::NetworkInterface
- AWS::EC2::SecurityGroup
- AWS::EC2::Subnet
- AWS::EC2::TransitGateway
- AWS::EC2::Volume
- AWS::EC2::VPC
- AWS::EC2::VPNConnection
- AWS::ECR::Repository
- AWS::ECS::Cluster
- AWS::ECS::Service
- AWS::ECS::TaskDefinition
- AWS::EFS::AccessPoint
- AWS::EFS::FileSystem
- AWS::EKS::Cluster
- AWS::ElasticBeanstalk::Environment
- AWS::ElasticLoadBalancing::LoadBalancer
- AWS::ElasticLoadBalancingV2::LoadBalancer
- AWS::Elasticsearch::Domain
- AWS::EMR::Cluster
- AWS::IAM::Group
- AWS::IAM::Policy
- AWS::IAM::Role
- AWS::IAM::User
- AWS::Kinesis::Stream
- AWS::KMS::Key
- AWS::Lambda::Function
- AWS::NetworkFirewall::FirewallPolicy
- AWS::OpenSearch::Domain
- AWS::RDS::DBCluster
- AWS::RDS::DBClusterSnapshot
- AWS::RDS::DBInstance
- AWS::RDS::DBSnapshot
- AWS::RDS::EventSubscription
- AWS::Redshift::Cluster
- AWS::S3::Bucket
- AWS::SageMaker::NotebookInstance
- AWS::SecretsManager::Secret
- AWS::SNS::Topic
- AWS::SQS::Queue
- AWS::SSM::AssociationCompliance
- AWS::SSM::Document
- AWS::SSM::PatchCompliance
- AWS::WAFRegional::Rule
- AWS::WAFRegional::RuleGroup
- AWS::WAFRegional::WebACL
- AWS::WAF::Rule

- `AWS::WAF::RuleGroup`
- `AWS::WAF::WebACL`

AWS Foundational Security Best Practices controls

The AWS Foundational Security Best Practices standard contains the following controls. For each control, the information includes the following information.

- The category that the control applies to. For descriptions of the categories, see [the section called "Control categories" \(p. 599\)](#).
- The severity
- The applicable resource that the control evaluates. We also list dependent resources for which you should enable AWS Config recording for the control to work.
- The required AWS Config rule, and any specific parameter values set by AWS Security Hub
- Remediation steps

Note that gaps in the control numbers indicate controls that are not yet released.

Controls categorized by service

[AWS Certificate Manager \(p. 460\)](#)

[Amazon API Gateway \(p. 461\)](#)

[Amazon EC2 Auto Scaling \(p. 464\)](#)

[AWS CloudFormation \(p. 469\)](#)

[Amazon CloudFront \(p. 470\)](#)

[AWS CloudTrail \(p. 475\)](#)

[AWS CodeBuild \(p. 479\)](#)

[AWS Config \(p. 482\)](#)

[AWS Database Migration Service \(p. 483\)](#)

[Amazon DynamoDB \(p. 484\)](#)

[Amazon EC2 \(p. 485\)](#)

[Amazon ECR \(p. 503\)](#)

[Amazon ECS \(p. 503\)](#)

[Amazon EFS \(p. 509\)](#)

[Amazon EKS \(p. 512\)](#)

[AWS Elastic Beanstalk \(p. 512\)](#)

[Elastic Load Balancing \(p. 514\)](#)

[Amazon EMR \(p. 523\)](#)

[Amazon ES \(p. 523\)](#)

[Amazon GuardDuty \(p. 528\)](#)

[AWS Identity and Access Management \(p. 529\)](#)

[Amazon Kinesis \(p. 537\)](#)

[AWS Key Management Service \(p. 537\)](#)

[AWS Lambda \(p. 540\)](#)

[AWS Network Firewall \(p. 543\)](#)

[Amazon OpenSearch Service \(p. 545\)](#)

[Amazon Relational Database Service \(p. 550\)](#)

[Amazon Redshift \(p. 569\)](#)

[Amazon Simple Storage Service \(p. 574\)](#)

[Amazon SageMaker \(p. 584\)](#)

[AWS Secrets Manager \(p. 585\)](#)

[Amazon Simple Notification Service \(p. 588\)](#)

[Amazon Simple Queue Service \(p. 590\)](#)

[Amazon EC2 Systems Manager \(p. 590\)](#)

[AWS WAF \(p. 594\)](#)

[ACM.1] Imported and ACM-issued certificates should be renewed after a specified time period

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::ACM::Certificate

AWS Config rule: [acm-certificate-expiration-check](#)

Schedule type: Change triggered

Parameters:

- `daysToExpiration`: 30

This control checks whether ACM certificates in your account are marked for expiration within 30 days. It checks both imported certificates and certificates provided by AWS Certificate Manager.

ACM can automatically renew certificates that use DNS validation. For certificates that use email validation, you must respond to a domain validation email. ACM also does not automatically renew certificates that you import. You must renew imported certificates manually.

For more information about managed renewal for ACM certificates, see [Managed renewal for ACM certificates](#) in the *AWS Certificate Manager User Guide*.

Note

This control is not supported in the following Regions.

- Africa (Cape Town)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)

Remediation

ACM provides managed renewal for your Amazon-issued SSL/TLS certificates. This means that ACM either renews your certificates automatically (if you use DNS validation), or it sends you email notices when the certificate expiration approaches. These services are provided for both public and private ACM certificates.

For domains validated by email

When a certificate is 45 days from expiration, ACM sends to the domain owner an email for each domain name. To validate the domains and complete the renewal, you must respond to the email notifications.

For more information, see [Renewal for domains validated by email](#) in the *AWS Certificate Manager User Guide*.

For domains validated by DNS

ACM automatically renews certificates that use DNS validation. 60 days before the expiration, ACM verifies that the certificate can be renewed.

If it cannot validate a domain name, then ACM sends a notification that manual validation is required. It sends these notifications 45 days, 30 days, 7 days, and 1 day before the expiration.

For more information, see [Renewal for domains validated by DNS](#) in the *AWS Certificate Manager User Guide*.

[APIGateway.1] API Gateway REST and WebSocket API logging should be enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::ApiGateway::Stage, AWS::ApiGatewayV2::Stage

AWS Config rule: [api-gw-execution-logging-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether all stages of an Amazon API Gateway REST or WebSocket API have logging enabled. The control fails if logging is not enabled for all methods of a stage or if `loggingLevel` is neither `ERROR` nor `INFO`.

API Gateway REST or WebSocket API stages should have relevant logs enabled. API Gateway REST and WebSocket API execution logging provides detailed records of requests made to API Gateway REST and WebSocket API stages. The stages include API integration backend responses, Lambda authorizer responses, and the `requestId` for AWS integration endpoints.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Europe (Milan)

Remediation

To enable logging for REST and WebSocket API operations, see [Set up CloudWatch API logging using the API Gateway console](#) in the *API Gateway Developer Guide*.

[APIGateway.2] API Gateway REST API stages should be configured to use SSL certificates for backend authentication

Category: Protect > Data protection

Severity: Medium

Resource type: AWS::ApiGateway::Stage

AWS Config rule: [api-gw-ssl-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon API Gateway REST API stages have SSL certificates configured. Backend systems use these certificates to authenticate that incoming requests are from API Gateway.

API Gateway REST API stages should be configured with SSL certificates to allow backend systems to authenticate that requests originate from API Gateway.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For detailed instructions on how to generate and configure API Gateway REST API SSL certificates, see [Generate and configure an SSL certificate for backend authentication](#) in the *API Gateway Developer Guide*.

[APIGateway.3] API Gateway REST API stages should have AWS X-Ray tracing enabled

Category: Detect > Detection services

Severity: Low

Resource type: AWS::ApiGateway::Stage

AWS Config rule: [api-gw-xray-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether AWS X-Ray active tracing is enabled for your Amazon API Gateway REST API stages.

X-Ray active tracing enables a more rapid response to performance changes in the underlying infrastructure. Changes in performance could result in a lack of availability of the API. X-Ray active tracing provides real-time metrics of user requests that flow through your API Gateway REST API operations and connected services.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For detailed instructions on how to enable X-Ray active tracing for API Gateway REST API operations, see [Amazon API Gateway active tracing support for AWS X-Ray](#) in the *AWS X-Ray Developer Guide*.

[APIGateway.4] API Gateway should be associated with an AWS WAF web ACL

Category: Protect > Protective services

Severity: Medium

Resource type: AWS::ApiGateway::Stage

AWS Configrule: [api-gw-associated-with-waf](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an API Gateway stage uses an AWS WAF web access control list (ACL). This control fails if an AWS WAF web ACL is not attached to a REST API Gateway stage.

AWS WAF is a web application firewall that helps protect web applications and APIs from attacks. It enables you to configure an ACL, which is a set of rules that allow, block, or count web requests based on customizable web security rules and conditions that you define. Ensure that your API Gateway stage is associated with an AWS WAF web ACL to help protect it from malicious attacks.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information on how to use the API Gateway console to associate an AWS WAF Regional web ACL with an existing API Gateway API stage, see [Using AWS WAF to protect your APIs](#) in the *API Gateway Developer Guide*.

[APIGateway.5] API Gateway REST API cache data should be encrypted at rest

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::ApiGateway::Stage

AWS Config rule: `api-gw-cache-encrypted` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether all methods in API Gateway REST API stages that have cache enabled are encrypted. The control fails if any method in an API Gateway REST API stage is configured to cache and the cache is not encrypted.

Encrypting data at rest reduces the risk of data stored on disk being accessed by a user not authenticated to AWS. It adds another set of access controls to limit unauthorized users ability access the data. For example, API permissions are required to decrypt the data before it can be read.

API Gateway REST API caches should be encrypted at rest for an added layer of security.

Remediation

To remediate this control, configure the stage to encrypt the cache data.

To configure API caching for a given stage

1. Open the API Gateway console at <https://console.aws.amazon.com/apigateway/>.
2. Choose the API.
3. Choose **Stages**.
4. In the **Stages** list for the API, choose the stage to add caching to.
5. Choose **Settings**.
6. Choose **Enable API cache**.
7. Update the desired settings, then select **Encrypt cache data**.
8. Choose **Save Changes**.

[AutoScaling.1] Auto Scaling groups associated with a load balancer should use load balancer health checks

Category: Identify > Inventory

Severity: Low

Resource type: AWS::AutoScaling::AutoScalingGroup

AWS Config rule: `autoscaling-group-elb-healthcheck-required`

Schedule type: Change triggered

Parameters: None

This control checks whether your Auto Scaling groups that are associated with a load balancer are using Elastic Load Balancing health checks.

This ensures that the group can determine an instance's health based on additional tests provided by the load balancer. Using Elastic Load Balancing health checks can help support the availability of applications that use EC2 Auto Scaling groups.

Remediation

To remediate this issue, update your Auto Scaling groups to use Elastic Load Balancing health checks.

To enable Elastic Load Balancing health checks

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
3. Select the check box for your group.
4. Choose **Edit**.
5. Under **Health checks**, for **Health check type**, choose **ELB**.
6. For **Health check grace period**, enter **300**.
7. At the bottom of the page, choose **Update**.

For more information on using a load balancer with an Auto Scaling group, see the [AWS Auto Scaling User Guide](#).

[AutoScaling.2] Amazon EC2 Auto Scaling group should cover multiple Availability Zones

Category: Recover > Resilience > High Availability

Severity: Medium

Resource type: AWS::AutoScaling::AutoScalingGroup

AWS Config rule: `autoscaling-multiple-az`

Schedule type: Change triggered

Parameters: None

This control checks whether an Auto Scaling group spans multiple Availability Zones. The control fails if an Auto Scaling group does not span multiple availability zones.

Amazon EC2 Auto Scaling groups can be configured to use multiple Availability Zones. An Auto Scaling group with a single Availability Zone is preferred in some use cases, such as batch-jobs or when inter-AZ transfer costs need to be kept to a minimum. However, an Auto Scaling group that does not span multiple Availability Zones will not launch instances in another Availability Zone to compensate if the configured single Availability Zone becomes unavailable.

Remediation

For information on how to add Availability Zones to an existing auto scaling group, see [Availability zones](#) in the *Amazon EC2 Auto Scaling User Guide*.

[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::AutoScaling::LaunchConfiguration`

AWS Config rule: [autoscaling-launchconfig-requires-imdsv2](#)

Schedule type: Change triggered

Parameters: None

This control checks whether IMDSv2 is enabled on all instances launched by Amazon EC2 Auto Scaling groups. The control fails if the Instance Metadata Service (IMDS) version is not included in the launch configuration or if both IMDSv1 and IMDSv2 are enabled.

IMDS provides data about your instance that you can use to configure or manage the running instance.

Version 2 of the IMDS adds new protections that weren't available in IMDSv1 to further safeguard your EC2 instances.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

An Auto Scaling group is associated with one launch configuration at a time. You cannot modify a launch configuration after you create it. To change the launch configuration for an Auto Scaling group, use an existing launch configuration as the basis for a new launch configuration with IMDSv2 enabled. For more information, see [Configure instance metadata options for new instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::AutoScaling::LaunchConfiguration`

AWS Config rule: [autoscaling-launch-config-hop-limit](#)

Schedule type: Change triggered

Parameters: None

This control checks the number of network hops that a metadata token can travel. The control fails if the metadata response hop limit is greater than 1.

The Instance Metadata Service (IMDS) provides metadata information about an Amazon EC2 instance and is useful for application configuration. Restricting the HTTP `PUT` response for the metadata service to only the EC2 instance protects the IMDS from unauthorized use.

The Time To Live (TTL) field in the IP packet is reduced by one on every hop. This reduction can be used to ensure that the packet does not travel outside EC2. IMDSv2 protects EC2 instances that may have

been misconfigured as open routers, layer 3 firewalls, VPNs, tunnels, or NAT devices, which prevents unauthorized users from retrieving metadata. With IMDSv2, the PUT response that contains the secret token cannot travel outside the instance because the default metadata response hop limit is set to 1. However, if this value is greater than 1, the token can leave the EC2 instance.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For detailed instructions on how to modify the metadata response hop limit for an existing launch configuration, see [Modify instance metadata options for existing instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

[AutoScaling.5] Amazon EC2 instances launched using Auto Scaling group launch configurations should not have Public IP addresses

Category: Protect > Secure network configuration

Severity: High

Resource type: AWS::AutoScaling::LaunchConfiguration

AWS Config rule: `autoscaling-launch-config-public-ip-disabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Auto Scaling group's associated launch configuration assigns a [public IP address](#) to the group's instances. The control fails if the associated launch configuration assigns a public IP address.

Amazon EC2 instances in an Auto Scaling group launch configuration should not have an associated public IP address, except for in limited edge cases. Amazon EC2 instances should only be accessible from behind a load balancer instead of being directly exposed to the internet.

Note

This control is only supported in US East (N. Virginia).

Remediation

An Auto Scaling group is associated with one launch configuration at a time. You cannot modify a launch configuration after you have create it. To change the launch configuration for an Auto Scaling group, use an existing launch configuration as the basis for a new launch configuration. Then, update the Auto Scaling group to use the new launch configuration as described in steps below.

After you change the launch configuration for an Auto Scaling group, any new instances are launched with the new configuration options. Existing instances are not affected. To update existing instances, either terminate them so that they are replaced by your Auto Scaling group, or allow automatic scaling to gradually replace older instances with newer instances based on your [termination policies](#).

To enable Elastic Load Balancing health checks

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Auto Scaling**, choose **Launch Configurations**.
3. Select the launch configuration and choose **Actions**, then **Copy launch configuration**. This sets up a new launch configuration with the same options as the original, but with "Copy" added to the name.
4. On the **Create Launch Configuration** page, expand **Advanced details** under **Additional configuration - optional**.
5. Under **IP address type**, choose **Do not assign a public IP address to any instances**.
6. When you have finished, Choose **Create launch configuration**.
7. On the navigation pane, under **Auto Scaling**, choose **Auto Scaling Groups**.
8. Select the check box next to the Auto Scaling group.
9. A split pane opens up in the bottom part of the page, showing information about the group that's selected.
10. On the **Details** tab, choose **Launch configuration, Edit**.
11. For **Launch configuration**, select the new launch configuration.
12. When you have finished changing your launch configuration, choose **Update**.

[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones

Category: Recover > Resilience > High Availability

Severity: Medium

Resource type: AWS::AutoScaling::AutoScalingGroup

AWS Config rule: [autoscaling-multiple-instance-types](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon EC2 Auto Scaling group uses multiple instance types. The control fails if the Auto Scaling group has only one instance type defined.

You can enhance availability by deploying your application across multiple instance types running in multiple Availability Zones. Security Hub recommends using multiple instance types so that the Auto Scaling group can launch another instance type if there is insufficient instance capacity in your chosen Availability Zones.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For detailed instructions on how to modify the metadata response hop limit for an existing launch configuration, see [Modify instance metadata options for existing instances](#) in the *Amazon EC2 User Guide*

for *Linux Instances* and [Modify instance metadata options for existing instances](#) in the *Amazon EC2 User Guide for Windows Instances*.

[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: AWS::CloudFormation::Stack

AWS Config rule: [cloudformation-stack-notification-check](#)

Schedule type: Change triggered

Parameters:

- `SNSTopic1`: 30
- `SNSTopic2`: 30
- `SNSTopic3`: 30
- `SNSTopic4`: 30
- `SNSTopic5`: 30
- (Optional): `SNS topic ARN`: 30

This control checks whether an Amazon Simple Notification Service notification is integrated with a CloudFormation stack. The control fails for a CloudFormation stack if there is no SNS notification associated with it.

Configuring an SNS notification with your CloudFormation stack helps immediately notify stakeholders of any events or changes occurring with the stack.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Middle East (Bahrain)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information about how to update a CloudFormation stack, see [AWS CloudFormation stack updates](#) in the AWS CloudFormation User Guide.

[CloudFront.1] CloudFront distributions should have a default root object configured

Category: Protect > Secure access management > Resources not publicly accessible

Severity: Critical

Resource type: AWS::CloudFront::Distribution

AWS Config rule: `cloudfront-default-root-object-configured`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution is configured to return a specific object that is the default root object. The control fails if the CloudFront distribution does not have a default root object configured.

A user might sometimes request the distribution's root URL instead of an object in the distribution. When this happens, specifying a default root object can help you to avoid exposing the contents of your web distribution.

Note

This control is only supported in US East (N. Virginia).

Remediation

For detailed instructions on how to specify a default root object for your distribution, see [How to specify a default root object](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.2] CloudFront distributions should have origin access identity enabled

Category: Protect > Secure access management > Resource policy configuration

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: `cloudfront-origin-access-identity-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution with Amazon S3 Origin type has Origin Access Identity (OAI) configured. The control fails if OAI is not configured.

CloudFront OAI prevents users from accessing S3 bucket content directly. When users access an S3 bucket directly, they effectively bypass the CloudFront distribution and any permissions that are applied to the underlying S3 bucket content.

Note

This control is only supported in US East (N. Virginia).

Remediation

For detailed remediation instructions, see [Creating a CloudFront OAI and adding it to your distribution](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.3] CloudFront distributions should require encryption in transit

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: [cloudfront-viewer-policy-https](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution requires viewers to use HTTPS directly or whether it uses redirection. The control fails if `ViewerProtocolPolicy` is set to `allow-all` for `defaultCacheBehavior` or for `cacheBehaviors`.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS.

Note

This control is only supported in US East (N. Virginia).

Remediation

For detailed remediation instructions, see [Requiring HTTPS for communication between viewers and CloudFront](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.4] CloudFront distributions should have origin failover configured

Category: Recover > Resilience > High availability

Severity: Low

Resource type: AWS::CloudFront::Distribution

AWS Config rule: [cloudfront-origin-failover-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon CloudFront distribution is configured with an origin group that has two or more origins.

CloudFront origin failover can increase availability. Origin failover automatically redirects traffic to a secondary origin if the primary origin is unavailable or if it returns specific HTTP response status codes.

Note

This control is only supported in US East (N. Virginia).

Remediation

For detailed remediation instructions, see [Creating an origin group](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.5] CloudFront distributions should have logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: [cloudfront-accesslogs-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether server access logging is enabled on CloudFront distributions. The control fails if access logging is not enabled for a distribution.

CloudFront access logs provide detailed information about every user request that CloudFront receives. Each log contains information such as the date and time the request was received, the IP address of the viewer that made the request, the source of the request, and the port number of the request from the viewer.

These logs are useful for applications such as security and access audits and forensics investigation. For additional guidance on how to analyze access logs, see [Querying Amazon CloudFront logs](#) in the *Amazon Athena User Guide*.

Note

This control is only supported in US East (N. Virginia).

Remediation

For information on how to configure access logging for a CloudFront distribution, see [Configuring and using standard logs \(access logs\)](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.6] CloudFront distributions should have AWS WAF enabled

Category: Protect > Protective services

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: [cloudfront-associated-with-waf](#)

Schedule type: Change triggered

Parameters: None

This control checks whether CloudFront distributions are associated with either AWS WAF or AWS WAFv2 web ACLs. The control fails if the distribution is not associated with a web ACL.

AWS WAF is a web application firewall that helps protect web applications and APIs from attacks. It allows you to configure a set of rules, called a web access control list (web ACL), that allow, block, or count web requests based on customizable web security rules and conditions that you define. Ensure your CloudFront distribution is associated with an AWS WAF web ACL to help protect it from malicious attacks.

Note

This control is only supported in US East (N. Virginia).

Remediation

For information on how to associate a web ACL with a CloudFront distribution, see [Using AWS WAF to control access to your content](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.7] CloudFront distributions should use custom SSL/TLS certificates

Category: Protect > Data protection > Encryption of data-in-transit

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: `cloudfront-custom-ssl-certificate`

Schedule type: Change triggered

Parameters: None

This control checks whether CloudFront distributions are using the default SSL/TLS certificate CloudFront provides. This control passes if the CloudFront distribution uses a custom SSL/TLS certificate. This control fails if the CloudFront distribution uses the default SSL/TLS certificate.

Custom SSL/TLS allow your users to access content by using alternate domain names. You can store custom certificates in AWS Certificate Manager (recommended), or in IAM.

Note

This control is only supported in US East (N. Virginia).

Remediation

To add an alternate domain name using a custom SSL/TLS certificate for your CloudFront distributions, see [Adding an alternate domain name](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.8] CloudFront distributions should use SNI to serve HTTPS requests

Category: Protect > Secure network configuration

Severity: Low

Resource type: AWS::CloudFront::Distribution

AWS Config rule: `cloudfront-sni-enabled`

Schedule type: Change triggered

Parameters: None

This control checks if Amazon CloudFront distributions are using a custom SSL/TLS certificate and are configured to use SNI to serve HTTPS requests. This control fails if a custom SSL/TLS certificate is associated but the SSL/TLS support method is a dedicated IP address.

Server Name Indication (SNI) is an extension to the TLS protocol that is supported by browsers and clients released after 2010. If you configure CloudFront to serve HTTPS requests using SNI, CloudFront associates your alternate domain name with an IP address for each edge location. When a viewer submits an HTTPS request for your content, DNS routes the request to the IP address for the correct edge location. The IP address to your domain name is determined during the SSL/TLS handshake negotiation; the IP address isn't dedicated to your distribution.

Note

This control is only supported in US East (N. Virginia).

Remediation

To configure your CloudFront distributions to use SNI to serve HTTPS requests, see [Using SNI to Serve HTTPS Requests \(works for Most Clients\)](#) in the *CloudFront Developer Guide*.

[CloudFront.9] CloudFront distributions should encrypt traffic to custom origins

Category: Protect > Data protection > Encryption of data-in-transit

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: [cloudfront-traffic-to-origin-encrypted](#)

Schedule type: Change triggered

Parameters: None

This control checks if Amazon CloudFront distributions are encrypting traffic to custom origins. This control fails for a CloudFront distribution whose origin protocol policy allows 'http-only'. This control also fails if the distribution's origin protocol policy is 'match-viewer' while the viewer protocol policy is 'allow-all'.

HTTPS (TLS) can be used to help prevent eavesdropping or manipulation of network traffic. Only encrypted connections over HTTPS (TLS) should be allowed.

Note

This control is only supported in US East (N. Virginia).

Remediation

To update the Origin Protocol Policy to require encryption for your CloudFront connections, see [Requiring HTTPS for communication between CloudFront and your custom origin](#) in the *Amazon CloudFront Developer Guide*.

[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins

Category: Protect > Data protection > Encryption of data-in-transit

Severity: Medium

Resource type: AWS::CloudFront::Distribution

AWS Config rule: [cloudfront-no-deprecated-ssl-protocols](#)

Schedule type: Change triggered

Parameters: None

This control checks if Amazon CloudFront distributions are using deprecated SSL protocols for HTTPS communication between CloudFront edge locations and your custom origins. This control fails if a CloudFront distribution has a `CustomOriginConfig` where `OriginSslProtocols` includes `SSLv3`.

In 2015, the Internet Engineering Task Force (IETF) officially announced that SSL 3.0 should be deprecated due to the protocol being insufficiently secure. It is recommended that you use TLSv1.2 or later for HTTPS communication to your custom origins.

Note

This control is only supported in US East (N. Virginia).

Remediation

To update the Origin SSL Protocols for your CloudFront distributions, see [Requiring HTTPS for communication between CloudFront and your custom origin](#) in the *Amazon CloudFront Developer Guide*.

[CloudTrail.1] CloudTrail should be enabled and configured with at least one multi-Region trail that includes read and write management events

Category: Identify > Logging

Severity: High

Resource type: AWS account

AWS Config rule: [multi-region-cloudtrail-enabled](#)

Schedule type: Periodic

Parameters:

- `readWriteType`: ALL

This control checks that there is at least one multi-Region CloudTrail trail. It also checks that the `ExcludeManagementEventSources` parameter is empty for at least one of those trails.

AWS CloudTrail records AWS API calls for your account and delivers log files to you. The recorded information includes the following information.

- Identity of the API caller
- Time of the API call
- Source IP address of the API caller
- Request parameters
- Response elements returned by the AWS service

CloudTrail provides a history of AWS API calls for an account, including API calls made from the AWS Management Console, AWS SDKs, command line tools. The history also includes API calls from higher-level AWS services such as AWS CloudFormation.

The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing. Multi-Region trails also provide the following benefits.

- A multi-Region trail helps to detect unexpected activity occurring in otherwise unused Regions.
- A multi-Region trail ensures that global service event logging is enabled for a trail by default. Global service event logging records events generated by AWS global services.
- For a multi-Region trail, management events for all read and write operations ensure that CloudTrail records management operations on all of an AWS account's resources.

By default, CloudTrail trails that are created using the AWS Management Console are multi-Region trails.

Remediation

To remediate this issue, create a new multi-Region trail in CloudTrail.

To create a new trail in CloudTrail

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. If you haven't used CloudTrail before, choose **Get Started Now**.
3. Choose **Trails** and then choose **Create trail**.
4. Enter a name for the trail.

5. Under **Storage location**, do one of the following:
 - a. To create a new S3 bucket for CloudTrail logs, for **Create a new S3 bucket**, choose **Yes**, then enter a name for the new S3 bucket.
 - b. To use an existing S3 bucket, for **Create a new S3 bucket**, choose **No**, then select the S3 bucket to use.
6. Under **Additional settings**, choose **Advanced**. For **Enable log file validation**, select **Enabled**.
7. Choose **Create**.

To update an existing trail in CloudTrail

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. In the **Name** column, choose the name of the trail.
4. For **Management events**, choose **Edit**.
5. For **Read/Write events**, select **Management events**.
6. Under **API Activity**, select **Read** and **Write**.

[CloudTrail.2] CloudTrail should have encryption at rest enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::CloudTrail::Trail

AWS Config rule: [cloud-trail-encryption-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail is configured to use the server-side encryption (SSE) AWS KMS key encryption. The check passes if the `KmsKeyId` is defined.

For an added layer of security for your sensitive CloudTrail log files, you should use [server-side encryption with AWS KMS–managed keys \(SSE-KMS\)](#) for your CloudTrail log files for encryption at rest. Note that by default, the log files delivered by CloudTrail to your buckets are encrypted by [Amazon server-side encryption with Amazon S3–managed encryption keys \(SSE-S3\)](#).

Remediation

To remediate this issue, update your trail to enable SSE-KMS encryption for the log files.

To enable encryption for CloudTrail logs

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Choose the trail to update.
4. Under **General details**, choose **Edit**.
5. For **Log file SSE-KMS encryption**, select **Enabled**.
6. For **Create a new KMS key**, do one of the following:
 - To create a key, choose **New**. Then in **AWS KMS alias**, enter an alias for the key. The key is created in the same Region as the S3 bucket.

- To use an existing key, choose **Existing**, then from **AWS KMS alias**, choose the key.

The AWS KMS key and S3 bucket must be in the same Region.

7. Choose **Save**.

You might need to modify the policy for CloudTrail to successfully interact with your KMS key. For more information, see [Encrypting CloudTrail log files with AWS KMS–managed keys \(SSE-KMS\)](#) in the *AWS CloudTrail User Guide*.

[CloudTrail.4] Ensure CloudTrail log file validation is enabled

Category: Data protection > Data integrity

Severity: Low

Resource type: AWS::CloudTrail::Trail

AWS Config rule: `cloud-trail-log-file-validation-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether log file integrity validation is enabled on a CloudTrail trail.

CloudTrail log file validation creates a digitally signed digest file that contains a hash of each log that CloudTrail writes to Amazon S3. You can use these digest files to determine whether a log file was changed, deleted, or unchanged after CloudTrail delivered the log.

Security Hub recommends that you enable file validation on all trails. Log file validation provides additional integrity checks of CloudTrail logs.

For more information, see [Enabling validation and validating files](#) in the *AWS CloudTrail User Guide*.

Remediation

To remediate this issue, update your CloudTrail trail to enable log file validation.

To enable CloudTrail log file validation

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Under **Name**, choose the name of a trail to edit.
4. Under **General details**, choose **Edit**.
5. Under **Additional settings**, for **Log file validation**, choose **Enabled**.
6. Choose **Save changes**.

For more information, see [Validating CloudTrail log file integrity](#) in the *AWS CloudTrail User Guide*.

[CloudTrail.5] Ensure CloudTrail trails are integrated with Amazon CloudWatch Logs

Category: Identify > Logging

Severity: Low

Resource type: `AWS::CloudTrail::Trail`

AWS Config rule: `cloud-trail-cloud-watch-logs-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether CloudTrail trails are configured to send logs to CloudWatch Logs. The control fails if the `CloudWatchLogsLogGroupArn` property of the trail is empty.

CloudTrail records AWS API calls that are made in a given account. The recorded information includes the following:

- The identity of the API caller
- The time of the API call
- The source IP address of the API caller
- The request parameters
- The response elements returned by the AWS service

CloudTrail uses Amazon S3 for log file storage and delivery. You can capture CloudTrail logs in a specified S3 bucket for long-term analysis. To perform real-time analysis, you can configure CloudTrail to send logs to CloudWatch Logs.

For a trail that is enabled in all Regions in an account, CloudTrail sends log files from all of those Regions to a CloudWatch Logs log group.

Security Hub recommends that you send CloudTrail logs to CloudWatch Logs. Note that this recommendation is intended to ensure that account activity is captured, monitored, and has appropriately alarms. You can use CloudWatch Logs to set this up with your AWS services. This recommendation does not preclude the use of a different solution.

Sending CloudTrail logs to CloudWatch Logs facilitates real-time and historic activity logging based on user, API, resource, and IP address. You can use this approach to establish alarms and notifications for anomalous or sensitivity account activity.

Remediation

You can use the console to enable CloudTrail integration with CloudWatch Logs.

To enable CloudTrail integration with CloudWatch Logs

1. Open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails**.
3. Choose the trail that does not have a value for **CloudWatch Logs Log group**.
4. Under **CloudWatch Logs**, choose **Edit**.
5. Select **Enabled**.
6. For **Log group**, do one of the following:
 - To use the default log group, keep the name as is.
 - To use an existing log group, choose **Existing** and then enter the name of the log group to use.
 - To create a new log group, choose **New** and then enter a name for the log group to create.
7. For **IAM role**, do one of the following:
 - To use an existing role, choose **Existing** and then choose the role from the drop-down list.

- To create a new role, choose **New** and then enter a name for the role to create. The new role is assigned a policy that grants the necessary permissions.

To view the permissions granted to the role, expand **Policy document**.

8. Choose **Save changes**.

For more information, see [Configuring CloudWatch Logs monitoring with the console](#) in the *AWS CloudTrail User Guide*.

[CodeBuild.1] CodeBuild GitHub or Bitbucket source repository URLs should use OAuth

Category: Protect > Secure development

Severity: Critical

Resource type: AWS::CodeBuild::Project

AWS Config rule: [codebuild-project-source-repo-url-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the GitHub or Bitbucket source repository URL contains either personal access tokens or a user name and password.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Authentication credentials should never be stored or transmitted in clear text or appear in the repository URL. Instead of personal access tokens or user name and password, you should use OAuth to grant authorization for accessing GitHub or Bitbucket repositories. Using personal access tokens or a user name and password could expose your credentials to unintended data exposure and unauthorized access.

Remediation

You can update your CodeBuild project to use OAuth.

To remove basic authentication / (GitHub) Personal Access Token from CodeBuild project source

1. Open the CodeBuild console at <https://console.aws.amazon.com/codebuild/>.
2. Choose the build project that contains personal access tokens or a user name and password.
3. From **Edit**, choose **Source**.
4. Choose **Disconnect from GitHub / Bitbucket**.
5. Choose **Connect using OAuth**, then choose **Connect to GitHub / Bitbucket**.
6. When prompted, choose **authorize as appropriate**.
7. Reconfigure your repository URL and additional configuration settings, as needed.

8. Choose **Update source**.

For more information, refer to [CodeBuild use case-based samples](#) in the *AWS CodeBuild User Guide*.

[CodeBuild.2] CodeBuild project environment variables should not contain clear text credentials

Category: Protect > Secure development

Severity: Critical

Resource type: AWS::CodeBuild::Project

AWS Config rule: [codebuild-project-envvar-awscred-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the project contains the environment variables `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`.

Authentication credentials `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY` should never be stored in clear text, as this could lead to unintended data exposure and unauthorized access.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To remediate this issue, update your CodeBuild project to remove the environment variable.

To remove environment variables from a CodeBuild project

1. Open the CodeBuild console at <https://console.aws.amazon.com/codebuild/>.
2. Expand **Build**.
3. Choose **Build project**, and then choose the build project that contains plaintext credentials.
4. From **Edit**, choose **Environment**.
5. Expand **Additional configuration**.
6. Choose **Remove** next to the environment variables.
7. Choose **Update environment**.

To store sensitive values in the Amazon EC2 Systems Manager Parameter Store and then retrieve them from your build spec

1. Open the CodeBuild console at <https://console.aws.amazon.com/codebuild/>.
2. Expand **Build**.
3. Choose **Build project**, and then choose the build project that contains plaintext credentials.
4. From **Edit**, choose **Environment**.

5. Expand **Additional configuration** and scroll to **Environment variables**.
6. Follow [this tutorial](#) to create a Systems Manager parameter that contains your sensitive data.
7. After you create the parameter, copy the parameter name.
8. Back in the CodeBuild console, choose **Create environmental variable**.
9. Enter the name of your variable as it appears in your build spec.
10. For **Value**, paste the name of your parameter.
11. For **Type**, choose **Parameter**.
12. To remove your noncompliant environmental variable that contains plaintext credentials, choose **Remove**.
13. Choose **Update environment**.

For more information, see [Environment variables in build environments](#) in the *AWS CodeBuild User Guide*.

[CodeBuild.4] CodeBuild project environments should have a logging configuration

Category: Identify > Logging

Severity: Medium

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether a CodeBuild project environment has at least one log option, either to S3 or CloudWatch logs enabled. This control fails if a CodeBuild project environment does not have at least one log option enabled.

From a security perspective, logging is an important feature to enable for future forensics efforts in the case of any security incidents. Correlating anomalies in CodeBuild projects with threat detections can increase confidence in the accuracy of those threat detections.

Remediation

For more information on how to configure CodeBuild project log settings, see [Create a build project \(console\)](#) in the CodeBuild User Guide.

[CodeBuild.5] CodeBuild project environments should not have privileged mode enabled

Category: Protect > Secure Access Management

Severity: High

Resource type: `AWS::CodeBuild::Project`

AWS Config rule: `codebuild-project-environment-privileged-check`

Schedule type: Change triggered

Parameters: None

This control checks if an AWS CodeBuild project environment has privileged mode enabled. This control fails when an AWS CodeBuild project environment has privileged mode enabled.

By default, Docker containers do not allow access to any devices. Privileged mode grants a build project's Docker container access to all devices. Setting `privilegedMode` with value `true` enables running the Docker daemon inside a Docker container. The Docker daemon listens for Docker API requests and manages Docker objects such as images, containers, networks, and volumes. This parameter should only be set to `true` if the build project is used to build Docker images. Otherwise, this setting should be disabled to prevent unintended access to Docker APIs as well as the container's underlying hardware as unintended access to `privilegedMode` may risk malicious tampering or deletion of critical resources.

Remediation

For more information on how to configure CodeBuild project environment settings, see [Create a build project \(console\)](#) in the CodeBuild User Guide

[Config.1] AWS Config should be enabled

Category: Identify > Inventory

Severity: Medium

Resource type: AWS account

AWS Config rule: None

Schedule type: Periodic

Parameters: None

This control checks whether AWS Config is enabled in the account for the local Region and is recording all resources.

The AWS Config service performs configuration management of supported AWS resources in your account and delivers log files to you. The recorded information includes the configuration item (AWS resource), relationships between configuration items, and any configuration changes between resources.

Security Hub recommends that you enable AWS Config in all Regions. The AWS configuration item history that AWS Config captures enables security analysis, resource change tracking, and compliance auditing.

Note

Because Security Hub is a Regional service, the check performed for this control checks only the current Region for the account. It does not check all Regions.

To allow security checks against global resources in each Region, you also must record global resources. If you only record global resources in a single Region, then you can disable this control in all Regions except the Region where you record global resources.

To learn more, see [Getting started with AWS Config](#) in the *AWS Config Developer Guide*.

Remediation

After you enable AWS Config, configure it to record all resources.

To configure AWS Config settings

1. Open the AWS Config console at <https://console.aws.amazon.com/config/>.
2. Select the Region to configure AWS Config in.
3. If you haven't used AWS Config before, see [Getting Started](#) in the *AWS Config Developer Guide*.
4. Navigate to the Settings page from the menu, and do the following:

- Choose **Edit**.
 - Under **Resource types to record**, select **Record all resources supported in this region and Include global resources (e.g., AWS IAM resources)**.
 - Under **Data retention period**, choose the default retention period for AWS Config data, or specify a custom retention period.
 - Under **AWS Config role**, either choose **Create AWS Config service-linked role** or choose **Choose a role from your account** and then select the role to use.
 - Under **Amazon S3 bucket**, specify the bucket to use or create a bucket and optionally include a prefix.
 - Under **Amazon SNS topic**, select an Amazon SNS topic from your account or create one. For more information about Amazon SNS, see the [Amazon Simple Notification Service Getting Started Guide](#).
5. Choose **Save**.

For more information about using AWS Config from the AWS CLI, see [Turning on AWS Config](#) in the *AWS Config Developer Guide*.

You can also use an AWS CloudFormation template to automate this process. For more information, see the [AWS CloudFormation StackSets sample template](#) in the *AWS CloudFormation User Guide*.

[DMS.1] AWS Database Migration Service replication instances should not be public

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS::DMS::ReplicationInstance

AWS Config rule: `dms-replication-not-public`

Schedule type: Periodic

Parameters: None

This control checks whether AWS DMS replication instances are public. To do this, it examines the value of the `PubliclyAccessible` field.

A private replication instance has a private IP address that you cannot access outside of the replication network. A replication instance should have a private IP address when the source and target databases are in the same network. The network must also be connected to the replication instance's VPC using a VPN, AWS Direct Connect, or VPC peering. To learn more about public and private replication instances, see [Public and private replication instances](#) in the *AWS Database Migration Service User Guide*.

You should also ensure that access to your AWS DMS instance configuration is limited to only authorized users. To do this, restrict users' IAM permissions to modify AWS DMS settings and resources.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

Note that you cannot change the public access setting once a replication instance is created. It must be deleted and recreated.

To configure the AWS DMS replication instances setting to block public access

1. Open the AWS Database Migration Service console at <https://console.aws.amazon.com/dms/>.

2. Navigate to **Replication instances**, then delete the public instance. Choose the instance, choose **Actions**, then choose **delete**.
3. Choose **Create replication instance**. Provide the configuration details.
4. To disable public access, make sure that **Publicly accessible** is not selected.
5. Choose **Create**.

For more information, see the section on [Creating a replication instance](#) in the *AWS Database Migration Service User Guide*.

[DynamoDB.1] DynamoDB tables should automatically scale capacity with demand

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: AWS::DynamoDB::Table

AWS Config rule: [dynamodb-autoscaling-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether an Amazon DynamoDB table can scale its read and write capacity as needed. This control passes if the table uses either on-demand capacity mode or provisioned mode with auto scaling configured. Scaling capacity with demand avoids throttling exceptions, which helps to maintain availability of your applications.

DynamoDB tables in on-demand capacity mode are only limited by the DynamoDB throughput default table quotas. To raise these quotas, you can file a support ticket through [AWS Support](#).

DynamoDB tables in provisioned mode with auto scaling adjust the provisioned throughput capacity dynamically in response to traffic patterns. For additional information on DynamoDB request throttling, see [Request throttling and burst capacity](#) in the *Amazon DynamoDB Developer Guide*.

Note

This control is not supported in AWS GovCloud (US-East) or AWS GovCloud (US-West).

Remediation

For detailed instructions on enabling DynamoDB automatic scaling on existing tables in capacity mode, see [Enabling DynamoDB auto scaling on existing tables](#) in the *Amazon DynamoDB Developer Guide*.

[DynamoDB.2] DynamoDB tables should have point-in-time recovery enabled

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: AWS::DynamoDB::Table

AWS Config rule: [dynamodb-pitr-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether point-in-time recovery (PITR) is enabled for an Amazon DynamoDB table.

Backups help you to recover more quickly from a security incident. They also strengthen the resilience of your systems. DynamoDB point-in-time recovery automates backups for DynamoDB tables. It reduces the time to recover from accidental delete or write operations. DynamoDB tables that have PITR enabled can be restored to any point in time in the last 35 days.

Remediation

To remediate this issue, add point-in-time recovery to your DynamoDB table.

To enable DynamoDB point-in-time recovery for an existing table

1. Open the DynamoDB console at <https://console.aws.amazon.com/dynamodb/>.
2. Choose the table that you want to work with, and then choose **Backups**.
3. In the **Point-in-time Recovery** section, under **Status**, choose **Enable**.
4. Choose **Enable** again to confirm the change.

[DynamoDB.3] DynamoDB Accelerator (DAX) clusters should be encrypted at rest

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::DAX::Cluster

AWS Config rule: [dax-encryption-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether a DAX cluster is encrypted at rest.

Encrypting data at rest reduces the risk of data stored on disk being accessed by a user not authenticated to AWS. The encryption adds another set of access controls to limit the ability of unauthorized users to access to the data. For example, API permissions are required to decrypt the data before it can be read.

Remediation

You cannot enable or disable encryption at rest after a cluster is created. You must recreate the cluster in order to enable encryption at rest. For detailed instructions on how to create a DAX cluster with encryption at rest enabled, see [Enabling encryption at rest using the AWS Management Console](#) in the *Amazon DynamoDB Developer Guide*.

[EC2.1] Amazon EBS snapshots should not be public, determined by the availability to be restorable by anyone

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS account

AWS Config rule: [ebs-snapshot-public-restorable-check](#)

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic Block Store snapshots are not public. The control fails if Amazon EBS snapshots are restorable by anyone.

EBS snapshots are used to back up the data on your EBS volumes to Amazon S3 at a specific point in time. You can use the snapshots to restore previous states of EBS volumes. It is rarely acceptable to share a snapshot with the public. Typically the decision to share a snapshot publicly was made in error or without a complete understanding of the implications. This check helps ensure that all such sharing was fully planned and intentional.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

To remediate this issue, update your EBS snapshot to make it private instead of public.

To make a public EBS snapshot private

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Elastic Block Store**, choose **Snapshots** menu and then choose your public snapshot.
3. From **Actions**, choose **Modify permissions**.
4. Choose **Private**.
5. (Optional) Add the AWS account numbers of the authorized accounts to share your snapshot with and choose **Add Permission**.
6. Choose **Save**.

[EC2.2] The VPC default security group should not allow inbound and outbound traffic

Category: Protect > Secure network configuration

Severity: High

Resource type: AWS::EC2::SecurityGroup

AWS Config rule: [vpc-default-security-group-closed](#)

Schedule type: Change triggered

Parameters: None

This control checks that the default security group of a VPC does not allow inbound or outbound traffic.

The rules for the [default security group](#) allow all outbound and inbound traffic from network interfaces (and their associated instances) that are assigned to the same security group.

We do not recommend using the default security group. Because the default security group cannot be deleted, you should change the default security group rules setting to restrict inbound and outbound traffic. This prevents unintended traffic if the default security group is accidentally configured for resources such as EC2 instances.

Remediation

To remediate this issue, create new security groups and assign those security groups to your resources. To prevent the default security groups from being used, remove their inbound and outbound rules.

To create new security groups and assign them to your resources

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security groups**. View the default security groups details to see the resources that are assigned to them.
3. Create a set of least-privilege security groups for the resources. For details on how to create security groups, see [Creating a security group](#) in the *Amazon VPC User Guide*.
4. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
5. On the Amazon EC2 console, change the security group for the resources that use the default security groups to the least-privilege security group you created. See [Changing an instance's security groups](#) in the *Amazon VPC User Guide*.

After you assign the new security groups to the resources, remove the inbound and outbound rules from the default security groups. This ensures that the default security groups are not used.

To remove the rules from the default security group

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Security groups**.
3. Select a default security group and choose the **Inbound rules** tab. Choose **Edit inbound rules**. Then delete all inbound rules. Choose **Save rules**.
4. Repeat the previous step for each default security group.
5. Select a default security group and choose the **Outbound rule** tab. Choose **Edit outbound rules**. Then delete all outbound rules. Choose **Save rules**.
6. Repeat the previous step for each default security group.

For more information, see [Working with security groups](#) in the *Amazon VPC User Guide*.

[EC2.3] Attached EBS volumes should be encrypted at rest

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::EC2::Volume

AWS Config rule: [encrypted-volumes](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the EBS volumes that are in an attached state are encrypted. To pass this check, EBS volumes must be in use and encrypted. If the EBS volume is not attached, then it is not subject to this check.

For an added layer of security of your sensitive data in EBS volumes, you should enable EBS encryption at rest. Amazon EBS encryption offers a straightforward encryption solution for your EBS resources that doesn't require you to build, maintain, and secure your own key management infrastructure. It uses KMS keys when creating encrypted volumes and snapshots.

To learn more about Amazon EBS encryption, see [Amazon EBS encryption](#) in the *Amazon EC2 User Guide for Linux Instances*.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

There is no direct way to encrypt an existing unencrypted volume or snapshot. You can only encrypt a new volume or snapshot when you create it.

If you enabled encryption by default, Amazon EBS encrypts the resulting new volume or snapshot using your default key for Amazon EBS encryption. Even if you have not enabled encryption by default, you can enable encryption when you create an individual volume or snapshot. In both cases, you can override the default key for Amazon EBS encryption and choose a symmetric customer managed key.

For more information, see [Creating an Amazon EBS volume](#) and [Copying an Amazon EBS snapshot](#) in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.4] Stopped EC2 instances should be removed after a specified time period

Category: Identify > Inventory

Severity: Medium

Resource type: AWS::EC2::Instance

AWS Config rule: [ec2-stopped-instance](#)

Schedule type: Periodic

Parameters:

- `allowedDays`: 30

This control checks whether any EC2 instances have been stopped for more than the allowed number of days. An EC2 instance fails this check if it is stopped for longer than the maximum allowed time period, which by default is 30 days.

A failed finding indicates that an EC2 instance has not run for a significant period of time. This creates a security risk because the EC2 instance is not being actively maintained (analyzed, patched, updated). If it is later launched, the lack of proper maintenance could result in unexpected issues in your AWS environment. To safely maintain an EC2 instance over time in a nonrunning state, start it periodically for maintenance and then stop it after maintenance. Ideally this is an automated process.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

You can terminate an EC2 instance using either the console or the command line.

Before you terminate the EC2 instance, verify that you won't lose any data:

- Check that your Amazon EBS volumes will not be deleted on termination.
- Copy any data that you need from your EC2 instance store volumes to Amazon EBS or Amazon S3.

To terminate an EC2 instance (console)

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, under **Instances**, choose **Instances**.
3. Select the instance, and then choose **Actions**, **Instance State**, **Terminate**.
4. When prompted for confirmation, choose **Yes**, **Terminate**.

To terminate an EC2 instance (AWS CLI, Tools for Windows PowerShell)

Use one of the following commands. For more information about the command line interface, see [Accessing Amazon EC2](#) in the *Amazon EC2 User Guide for Linux Instances*.

- From the AWS CLI, use `terminate-instances`
- From the Tools for Windows PowerShell, use `Stop-EC2Instance`.

To learn more about terminating instances, see [Terminating an instance](#) in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.6] VPC flow logging should be enabled in all VPCs

Category: Identify > Logging

Severity: Medium

Resource type: AWS::EC2::VPC

AWS Config rule: `vpc-flow-logs-enabled`

Schedule type: Periodic

Parameters:

- `trafficType: REJECT`

This control checks whether Amazon VPC Flow Logs are found and enabled for VPCs. The traffic type is set to Reject.

With the VPC Flow Logs feature, you can capture information about the IP address traffic going to and from network interfaces in your VPC. After you create a flow log, you can view and retrieve its data in CloudWatch Logs. To reduce cost, you can also send your flow logs to Amazon S3.

Security Hub recommends that you enable flow logging for packet rejects for VPCs. Flow logs provide visibility into network traffic that traverses the VPC and can detect anomalous traffic or provide insight during security workflows.

By default, the record includes values for the different components of the IP address flow, including the source, destination, and protocol. For more information and descriptions of the log fields, see [VPC Flow Logs](#) in the *Amazon VPC User Guide*.

Remediation

To remediate this issue, enable VPC flow logging.

To enable VPC flow logging

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. Under **Virtual Private Cloud**, choose **Your VPCs**.
3. Select a VPC to update.
4. At the bottom of the page, choose **Flow Logs**.
5. Choose **Create flow log**.
6. For **Filter**, choose **Reject**.
7. For **Destination log group**, choose the log group to use.
8. For **IAM role**, choose the IAM role to use.
9. Choose **Create**.

[EC2.7] EBS default encryption should be enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS account

AWS Config rule: [ec2-ebs-encryption-by-default](#)

Schedule type: Periodic

Parameters: None

This control checks whether account-level encryption is enabled by default for Amazon Elastic Block Store (Amazon EBS). The control fails if the account level encryption is not enabled.

When encryption is enabled for your account, Amazon EBS volumes and snapshot copies are encrypted at rest. This adds an additional layer of protection for your data. For more information, see [Encryption by default](#) in the *Amazon EC2 User Guide for Linux Instances*.

Note that following instance types do not support encryption: R1, C1, and M1.

Remediation

You can use the Amazon EC2 console to enable default encryption for Amazon EBS volumes.

To configure the default encryption for Amazon EBS encryption for a Region

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. From the navigation pane, select **EC2 Dashboard**.
3. In the upper-right corner of the page, choose **Account Attributes, EBS encryption**.
4. Choose **Manage**.
5. Select **Enable**. You can keep the AWS managed key with the alias `alias/aws/ebs` created on your behalf as the default encryption key, or choose a symmetric customer managed key.
6. Choose **Update EBS encryption**.

[EC2.8] EC2 instances should use IMDSv2

Category: Protect > Network security

Severity: High

Resource type: AWS::EC2::Instance

AWS Config rule: [ec2-imdsv2-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether your EC2 instance metadata version is configured with Instance Metadata Service Version 2 (IMDSv2). The control passes if `HttpTokens` is set to `required` for IMDSv2. The control fails if `HttpTokens` is set to `optional`.

You use instance metadata to configure or manage the running instance. The IMDS provides access to temporary, frequently rotated credentials. These credentials remove the need to hard code or distribute

sensitive credentials to instances manually or programmatically. The IMDS is attached locally to every EC2 instance. It runs on a special "link local" IP address of 169.254.169.254. This IP address is only accessible by software that runs on the instance.

Version 2 of the IMDS adds new protections for the following types of vulnerabilities. These vulnerabilities could be used to try to access the IMDS.

- Open website application firewalls
- Open reverse proxies
- Server-side request forgery (SSRF) vulnerabilities
- Open Layer 3 firewalls and network address translation (NAT)

Security Hub recommends that you configure your EC2 instances with IMDSv2.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

To remediate an EC2 instance that is not configured with IMDSv2, you can require the use of IMDSv2.

To require IMDSv2 on an existing instance, when you request instance metadata, modify the Amazon EC2 metadata options. Follow the instructions in [Configuring instance metadata options for existing instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

To require the use of IMDSv2 on a new instance when you launch it, follow the instructions in [Configuring instance metadata options for new instances](#) in the *Amazon EC2 User Guide for Linux Instances*.

To configure your new EC2 instance with IMDSv2 from the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch instance** and then choose **Launch instance**.
3. In the **Configure Instance Details** step, under **Advanced Details**, for **Metadata version**, choose **V2 (token required)**.
4. Choose **Review and Launch**.

If your software uses IMDSv1, you can reconfigure your software to use IMDSv2. For details, see [Transitioning to using Instance Metadata Service Version 2](#) in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.9] EC2 instances should not have a public IP address

Category: Protect > Secure network configuration > Public IP addresses

Severity: High

Resource type: AWS::EC2::Instance

AWS Config rule: [ec2-instance-no-public-ip](#)

Schedule type: Change triggered

Parameters: None

This control checks whether EC2 instances have a public IP address. The control fails if the `publicIp` field is present in the EC2 instance configuration item. This control applies to IPv4 addresses only.

A public IPv4 address is an IP address that is reachable from the internet. If you launch your instance with a public IP address, then your EC2 instance is reachable from the internet. A private IPv4 address is an IP address that is not reachable from the internet. You can use private IPv4 addresses for communication between EC2 instances in the same VPC or in your connected private network.

IPv6 addresses are globally unique, and therefore are reachable from the internet. However, by default all subnets have the IPv6 addressing attribute set to false. For more information about IPv6, see [IP addressing in your VPC](#) in the *Amazon VPC User Guide*.

If you have a legitimate use case to maintain EC2 instances with public IP addresses, then you can suppress the findings from this control. For more information about front-end architecture options, see the [AWS Architecture Blog](#) or the [This Is My Architecture series](#).

Remediation

Use a non-default VPC so that your instance is not assigned a public IP address by default.

When you launch an EC2 instance into a default VPC, it is assigned a public IP address. When you launch an EC2 instance into a non-default VPC, the subnet configuration determines whether it receives a public IP address. The subnet has an attribute to determine if new EC2 instances in the subnet receive a public IP address from the public IPv4 address pool.

You cannot manually associate or disassociate an automatically-assigned public IP address from your EC2 instance. To control whether your EC2 instance receives a public IP address, do one of the following:

- Modify the public IP addressing attribute of your subnet. For more information, see [Modifying the public IPv4 addressing attribute for your subnet](#) in the *Amazon VPC User Guide*.
- Enable or disable the public IP addressing feature during launch. This overrides the subnet's public IP addressing attribute. For more information, see [Assign a public IPv4 address during instance launch](#) in the *Amazon EC2 User Guide for Linux Instances*.

For more information, see [Public IPv4 addresses and external DNS hostnames](#) in the *Amazon EC2 User Guide for Linux Instances*.

If your EC2 instance is associated with an Elastic IP address, then your EC2 instance is reachable from the internet. You can disassociate an Elastic IP address from an instance or network interface at any time.

To disassociate an Elastic IP address

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Elastic IPs**.
3. Select the Elastic IP address to disassociate.
4. From **Actions**, choose **Disassociate Elastic IP address**.
5. Choose **Disassociate**.

[EC2.10] Amazon EC2 should be configured to use VPC endpoints that are created for the Amazon EC2 service

Category: Protect - Secure network configuration > API private access

Severity: Medium

Resource type: AWS : : EC2 : : VPC

AWS Config rule: [service-vpc-endpoint-enabled](#)

Schedule type: Periodic

Parameters:

- `serviceName: ec2`

This control checks whether a service endpoint for Amazon EC2 is created for each VPC. The control fails if a VPC does not have a VPC endpoint created for the Amazon EC2 service.

This control evaluates resources in single account. It cannot describe resources that are outside of the account. Because AWS Config and Security Hub do not conduct cross-account checks, you will see `FAILED` findings for VPCs that are shared across accounts. Security Hub recommends that you suppress these `FAILED` findings.

To improve the security posture of your VPC, you can configure Amazon EC2 to use an interface VPC endpoint. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to access Amazon EC2 API operations privately. It restricts all network traffic between your VPC and Amazon EC2 to the Amazon network. Because endpoints are supported within the same Region only, you cannot create an endpoint between a VPC and a service in a different Region. This prevents unintended Amazon EC2 API calls to other Regions.

To learn more about creating VPC endpoints for Amazon EC2, see [Amazon EC2 and interface VPC endpoints](#) in the *Amazon EC2 User Guide for Linux Instances*.

Remediation

To remediate this issue, you can create an interface VPC endpoint to Amazon EC2.

To create an interface endpoint to Amazon EC2 from the Amazon VPC console

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Endpoints**.
3. Choose **Create Endpoint**.
4. For **Service category**, choose **AWS services**.
5. For **Service Name**, choose **com.amazonaws.<region>.ec2**.
6. For **Type**, choose **Interface**.
7. Complete the following information.
 - a. For **VPC**, select a VPC in which to create the endpoint.
 - b. For **Subnets**, select the subnets (Availability Zones) in which to create the endpoint network interfaces. Not all Availability Zones are supported for all AWS services.
 - c. To enable private DNS for the interface endpoint, select the check box for **Enable DNS Name**. This option is enabled by default.

To use the private DNS option, the following attributes of your VPC must be set to true:

- `enableDnsHostnames`
- `enableDnsSupport`

For more information, see [Viewing and updating DNS support for your VPC](#) in the *Amazon VPC User Guide*.

- d. For **Security group**, select the security groups to associate with the endpoint network interfaces.
- e. (Optional) Add or remove a tag. To add a tag, choose **Add tag** and do the following:
 - For **Key**, enter the tag name.
 - For **Value**, enter the tag value.

- f. To remove a tag, choose the delete button (x) to the right of the tag **Key** and **Value**.
8. Choose **Create endpoint**.

To create an interface VPC endpoint policy

You can attach a policy to your VPC endpoint to control access to the Amazon EC2 API. The policy specifies the following:

- The principal that can perform actions
- The actions that can be performed
- The resource on which the actions can be performed

For more details on creating a VPC endpoint policy, see [Amazon EC2 and interface VPC endpoints](#) in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.15] EC2 subnets should not automatically assign public IP addresses

Category: Protect > Network security

Severity: Medium

Resource type: AWS::EC2::Subnet

AWS Config rule: [subnet-auto-assign-public-ip-disabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the assignment of public IPs in Amazon Virtual Private Cloud (Amazon VPC) subnets have `MapPublicIpOnLaunch` set to `FALSE`. The control passes if the flag is set to `FALSE`.

All subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. Instances that are launched into subnets that have this attribute enabled have a public IP address assigned to their primary network interface.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can configure a subnet from the Amazon VPC console.

To configure a subnet to not assign public IP addresses

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, choose **Subnets**.
3. Select your subnet and then choose **Subnet Actions, Modify auto-assign IP settings**.
4. Clear the **Enable auto-assign public IPv4 address** check box and then choose **Save**.

[EC2.16] Unused network access control lists should be removed

Category: Prevent > Network security

Severity: Low

Resource type: AWS::EC2::NetworkACL

AWS Config rule: [vpc-network-acl-unused-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether there are any unused network access control lists (ACLs).

The control checks the item configuration of the resource `AWS::EC2::NetworkACL` and determines the relationships of the network ACL.

If the only relationship is the VPC of the network ACL, then the control fails.

If other relationships are listed, then the control passes.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For instructions on how to delete an unused network ACL, see [Deleting a network ACL](#) in the *Amazon VPC User Guide*.

[EC2.17] EC2 instances should not use multiple ENIs

Category: Network security

Severity: Low

Resource type: AWS::EC2::Instance

AWS Config rule: [ec2-instance-multiple-eni-check](#)

Schedule type: Change triggered

Parameters:

- `Adapterids` (Optional) – A list of network interface IDs that are attached to EC2 instances

This control checks whether an EC2 instance uses multiple Elastic Network Interfaces (ENIs) or Elastic Fabric Adapters (EFAs). This control passes if a single network adapter is used. The control includes an optional parameter list to identify the allowed ENIs.

Multiple ENIs can cause dual-homed instances, meaning instances that have multiple subnets. This can add network security complexity and introduce unintended network paths and access.

This control also fails if an Amazon EKS cluster that belongs to an Amazon EKS cluster has more than one ENI. If you need to use EC2 instances that have multiple ENIs as part of an Amazon EKS cluster, you can suppress those findings.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To remediate this issue, detach the additional ENIs.

To detach a network interface

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Under **Network & Security**, choose **Network Interfaces**.
3. Filter the list by the noncompliant instance IDs to see the associated ENIs.
4. Select the ENIs that you want to remove.
5. From the **Actions** menu, choose **Detach**.
6. If you see the prompt **Are you sure that you want to detach the following network interface?**, choose **Detach**.

[EC2.18] Security groups should only allow unrestricted incoming traffic for authorized ports

Category: Protect > Secure network configuration > Security group configuration

Severity: High

Resource type: AWS::EC2::SecurityGroup

AWS Config rule: [vpc-sg-open-only-to-authorized-ports](#)

Schedule type: Change triggered

Parameters:

- `authorizedTcpPorts` (Optional) – Comma-separated list of ports to which to allow unrestricted access. For example: '80, 443'. For this rule, the default values for `authorizedTcpPorts` are 80 and 443.

This control checks whether the security groups that are in use allow unrestricted incoming traffic. Optionally the rule checks whether the port numbers are listed in the `authorizedTcpPorts` parameter.

- If the security group rule port number allows unrestricted incoming traffic, but the port number is specified in `authorizedTcpPorts`, then the control passes. The default value for `authorizedTcpPorts` is 80, 443.
- If the security group rule port number allows unrestricted incoming traffic, but the port number is not specified in `authorizedTcpPorts` input parameter, then the control fails.
- If the parameter is not used, then the control fails for any security group that has an unrestricted inbound rule.

Security groups provide stateful filtering of ingress and egress network traffic to AWS. Security group rules should follow the principal of least privileged access. Unrestricted access (IP address with a /0 suffix) increases the opportunity for malicious activity such as hacking, denial-of-service attacks, and loss of data.

Unless a port is specifically allowed, the port should deny unrestricted access.

Note

This control is not supported in Asia Pacific (Osaka).

Remediation

For information on how to modify a security group, see [Add, remove, or update rules](#) in the *Amazon VPC User Guide*.

[EC2.19] Security groups should not allow unrestricted access to ports with high risk

Category: Protect > Restricted network access

Severity: Critical

Resource type: AWS::EC2::SecurityGroup

AWS Config rule: vpc-sg-restricted-common-ports (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether unrestricted incoming traffic for the security groups is accessible to the specified ports that have the highest risk. This control passes when none of the rules in a security group allow ingress traffic from 0.0.0.0/0 for those ports.

Unrestricted access (0.0.0.0/0) increases opportunities for malicious activity, such as hacking, denial-of-service attacks, and loss of data.

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. No security group should allow unrestricted ingress access to the following ports:

- 20, 21 (FTP)
- 22 (SSH)
- 23 (Telnet)
- 25 (SMTP)
- 110 (POP3)
- 135 (RPC)
- 143 (IMAP)
- 445 (CIFS)
- 1433, 1434 (MSSQL)
- 3000 (Go, Node.js, and Ruby web development frameworks)
- 3306 (mySQL)
- 3389 (RDP)
- 4333 (ahsp)
- 5000 (Python web development frameworks)
- 5432 (postgresql)

- 5500 (fcp-addr-srvr1)
- 5601 (OpenSearch Dashboards)
- 8080 (proxy)
- 8088 (legacy HTTP port)
- 8888 (alternative HTTP port)
- 9200 or 9300 (OpenSearch)

Remediation

For information on how to delete rules from a security group, see [Delete rules from a security group](#) in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.20] Both VPN tunnels for an AWS Site-to-Site VPN connection should be up

Category: Resilience > Recover > High availability

Severity: Medium

Resource type: AWS::EC2::VPNConnection

AWS Config rule: [vpc-vpn-2-tunnels-up](#)

Schedule type: Change triggered

Parameters: None

A VPN tunnel is an encrypted link where data can pass from the customer network to or from AWS within an AWS Site-to-Site VPN connection. Each VPN connection includes two VPN tunnels which you can simultaneously use for high availability. Ensuring that both VPN tunnels are up for a VPN connection is important for confirming a secure and highly available connection between an AWS VPC and your remote network.

This control checks that both VPN tunnels provided by AWS Site-to-Site VPN are in UP status. The control fails if one or both tunnels are in DOWN status.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Middle East (Bahrain)

Remediation

To modify VPN tunnel options, see [Modifying Site-to-Site VPN tunnel options](#) in the AWS Site-to-Site VPN User Guide.

[EC2.21] Network ACLs should not allow ingress from 0.0.0.0/0 to port 22 or port 3389

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::EC2::NetworkAcl`

AWS Config rule: `nacl-no-unrestricted-ssh-rdp`

Schedule type: Change triggered

Parameters: None

This control checks whether a network access control list (NACL) allows unrestricted access to the default ports for SSH/RDP ingress traffic. The rule fails if a NACL inbound entry allows a source CIDR block of '0.0.0.0/0' or ':::0' for ports 22 or 3389.

Access to remote server administration ports, such as port 22 (SSH) and port 3389 (RDP), should not be publicly accessible, as this may allow unintended access to resources within your VPC.

Remediation

For more information about NACLs, see [Network ACLs](#) in the VPC User Guide.

[EC2.22] Unused EC2 security groups should be removed

Category: Identify > Inventory

Severity: Medium

Resource type: `AWS::EC2::SecurityGroup`, `AWS::EC2::NetworkInterface`

AWS Config rule: `ec2-security-group-attached-to-eni-periodic`

Schedule type: Periodic

Parameters: None

This AWS control checks that security groups are attached to Amazon Elastic Compute Cloud (Amazon EC2) instances or to an elastic network interface. The control will fail if the security group is not associated with an Amazon EC2 instance or an elastic network interface.

Remediation

To create, assign and delete security groups, see [Security groups](#) in Amazon EC2 user guide.

[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests

Category: Protect > Secure network configuration

Severity: High

Resource type: `AWS::EC2::TransitGateway`

AWS Config rule: `ec2-transit-gateway-auto-vpc-attach-disabled`

Schedule type: Change triggered

Parameters: None

This control checks if EC2 Transit Gateways are automatically accepting shared VPC attachments. This control fails for a Transit Gateway that automatically accepts shared VPC attachment requests.

Turning on `AutoAcceptSharedAttachments` configures a Transit Gateway to automatically accept any cross-account VPC attachment requests without verifying the request or the account the attachment is originating from. To follow the best practices of authorization and authentication, we recommended turning off this feature to ensure that only authorized VPC attachment requests are accepted.

Note

This control is not supported in the following Regions:

- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Middle East (Bahrain)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information about how to modify a Transit Gateway, see [Modify a transit gateway](#) in the Amazon VPC Developer Guide.

[EC2.24] Paravirtual EC2 instance types should not be used

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: `AWS::EC2::Instance`

AWS Config rule: [ec2-paravirtual-instance-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the virtualization type of an EC2 instance is paravirtual. The control fails if the `virtualizationType` of the EC2 instance is set to `paravirtual`.

Linux Amazon Machine Images (AMIs) use one of two types of virtualization: paravirtual (PV) or hardware virtual machine (HVM). The main differences between PV and HVM AMIs are the way in which they boot and whether they can take advantage of special hardware extensions (CPU, network, and storage) for better performance.

Historically, PV guests had better performance than HVM guests in many cases, but because of enhancements in HVM virtualization and the availability of PV drivers for HVM AMIs, this is no longer true. For more information, see [Linux AMI virtualization types](#) in the Amazon EC2 User Guide for Linux Instances.

Note

This control is not supported in the following Regions:

- US East (Ohio)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Jakarta)
- Asia Pacific (Mumbai)

- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Canada (Central)
- China (Beijing)
- China (Ningxia)
- Europe (London)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Middle East (Bahrain)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information about how to update an EC2 instance to a new instance type, see [Change the instance type](#) in the *Amazon EC2 User Guide for Linux Instances*.

[EC2.27] Running EC2 Instances should not use key pairs

Category: Identify > Resource configuration

Severity: High

Resource type: AWS :: EC2 :: Instance

AWS Config rule: [ec2-no-amazon-key-pair](#)

Schedule type: Change triggered

Parameters: None

This control checks whether running EC2 instances are using key pairs. The control fails if a running EC2 instance uses a key pair.

As best practice, we recommend that you reduce the number of credentials in use whenever possible to minimize the risk of compromised credentials and unintended access. EC2 instances without key pairs can still be accessed using AWS Systems Manager Session Manager or browser-based SSH connection via [the AWS console](#). You can also access EC2 instances with a password, but we do not recommend this because it involves credentials similar to key pairs.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To delete a key pair, see [Delete your public key on Amazon EC2](#) in the *Amazon EC2 User Guide for Linux Instances*.

[ECR.1] ECR private repositories should have image scanning configured

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: AWS::ECR::Repository

AWS Config rule: [ecr-private-image-scanning-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether a private ECR repository has image scanning configured. This control fails if a private ECR repository doesn't have image scanning configured.

ECR image scanning helps in identifying software vulnerabilities in your container images. ECR uses the Common Vulnerabilities and Exposures (CVEs) database from the [open-source Clair project](#) and provides a list of scan findings. Enabling image scanning on ECR repositories adds a layer of verification for the integrity and safety of the images being stored.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To configure image scanning for an ECR repository, see [Image scanning](#) in the *Amazon Elastic Container Registry User Guide*.

[ECR.2] ECR private repositories should have tag immutability configured

Category: Identify > Inventory > Tagging

Severity: Medium

Resource type: AWS::ECR::Repository

AWS Config rule: [ecr-private-tag-immutability-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether a private ECR repository has tag immutability enabled. This control fails if a private ECR repository has tag immutability disabled. This rule passes if tag immutability is enabled and has the value `IMMUTABLE`.

Amazon ECR Tag Immutability enables customers to rely on the descriptive tags of an image as a reliable mechanism to track and uniquely identify images. An immutable tag is static, which means each tag refers to a unique image. This improves reliability and scalability as the use of a static tag will always result in the same image being deployed. When configured, tag immutability prevents the tags from being overridden, which reduces the attack surface.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To configure image scanning for an ECR repository, see [Image scanning](#) in the *Amazon Elastic Container Registry User Guide*.

[ECR.3] ECR repositories should have at least one lifecycle policy configured

Category: Identify > Resource configuration

Severity: Medium

Resource type: AWS::ECR::Repository

AWS Config rule: `ecr-private-lifecycle-policy-configured`

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon ECR repository has at least one lifecycle policy configured. This control fails if an ECR repository does not have any lifecycle policies configured.

Amazon ECR lifecycle policies enable you to specify the lifecycle management of images in a repository. By configuring lifecycle policies, you can automate the cleanup of unused images and the expiration of images based on age or count. Automating these tasks can help you avoid unintentionally using outdated images in your repository.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To configure a lifecycle policy, see [Creating a lifecycle policy preview](#) in the *Amazon Elastic Container Registry User Guide*.

[ECS.1] Amazon ECS task definitions should have secure networking modes and user definitions

Category: Protect > Secure access management

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Config rule: [ecs-task-definition-user-for-host-mode-check](#)

Schedule type: Change triggered

Parameters:

- `SkipInactiveTaskDefinitions: true`

This control checks whether an active Amazon ECS task definition that has host networking mode also has `privileged` or `user` container definitions. The control fails for task definitions that have host network mode and container definitions where `privileged=false` or is empty and `user=root` or is empty.

If a task definition has elevated privileges, it is because the customer has specifically opted in to that configuration. This control checks for unexpected privilege escalation when a task definition has host networking enabled but the customer has not opted in to elevated privileges.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information on how to update a task definition, see [Updating a task definition](#) in the *Amazon Elastic Container Service Developer Guide*.

Note that when you update a task definition, it does not update running tasks that were launched from the previous task definition. To update a running task, you must redeploy the task with the new task definition.

[ECS.2] Amazon ECS services should not have public IP addresses assigned to them automatically

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: High

Resource type: `AWS::ECS::Service`

AWS Config rule: `ecs-service-assign-public-ip-disabled` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters:

- `exemptEcsServiceArns` (Optional). Security Hub does not populate this parameter. Comma-separated list of ARNs of Amazon ECS services that are exempt from this rule.

This rule is **COMPLIANT** if an Amazon ECS service has `AssignPublicIP` set to **ENABLED** and is specified in this parameter list.

This rule is **NON_COMPLIANT** if an Amazon ECS service has `AssignPublicIP` set to **ENABLED** and is not specified in this parameter list.

This control checks whether Amazon ECS services are configured to automatically assign public IP addresses. This control fails if `AssignPublicIP` is **ENABLED**. This control passes if `AssignPublicIP` is **DISABLED**.

A public IP address is an IP address that is reachable from the internet. If you launch your Amazon ECS instances with a public IP address, then your Amazon ECS instances are reachable from the internet. Amazon ECS services should not be publicly accessible, as this may allow unintended access to your container application servers.

Note

This control is not supported in the Asia Pacific (Osaka) Region.

Remediation

To disable automatic public IP assignment, see [To configure VPC and security group settings for your service](#) in the *Amazon Elastic Container Service Developer Guide*.

[ECS.3] ECS task definitions should not share the host's process namespace

Category: Identify > Resource configuration

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Configrule: [ecs-task-definition-pid-mode-check](#)

Schedule type: Change triggered

Parameters: None

This control checks if Amazon ECS task definitions are configured to share a host's process namespace with its containers. The control fails if the task definition shares the host's process namespace with the containers running on it.

A process ID (PID) namespace provides separation between processes. It prevents system processes from being visible, and allows PIDs to be reused, including PID 1. If the host's PID namespace is shared with containers, it would allow containers to see all of the processes on the host system. This reduces the benefit of process level isolation between the host and the containers. These circumstances could lead to unauthorized access to processes on the host itself, including the ability to manipulate and terminate them. Customers shouldn't share the host's process namespace with containers running on it.

Remediation

To configure the `pidMode` on a task definition, see [Task definition parameters](#) in the *Amazon Elastic Container Service Developer Guide*.

[ECS.4] ECS containers should run as non-privileged

Category: Protect > Secure access management > Root user access restrictions

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Configrule: [ecs-containers-nonprivileged](#)

Schedule type: Change triggered

Parameters: None

This control checks if the `privileged` parameter in the container definition of Amazon ECS Task Definitions is set to `true`. The control fails if this parameter is equal to `true`.

We recommend that you remove elevated privileges from your ECS task definitions. When the `privilege` parameter is `true`, the container is given elevated privileges on the host container instance (similar to the root user).

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To configure the `privileged` parameter on a task definition, see [Advanced container definition parameters](#) in the Amazon Elastic Container Service Developer Guide.

[ECS.5] ECS containers should be limited to read-only access to root filesystems

Category: Protect > Secure access management

Severity: High

Resource type: `AWS::ECS::TaskDefinition`

AWS Configrule: [ecs-containers-readonly-access](#)

Schedule type: Change triggered

Parameters: None

This control checks if ECS containers are limited to read-only access to mounted root filesystems. This control fails if the `ReadOnlyRootFilesystem` parameter in the container definition of ECS task definitions is set to `false`.

Enabling this option reduces security attack vectors since the container instance's filesystem cannot be tampered with or written to unless it has explicit read-write permissions on its filesystem folder and directories. This control also adheres to the principle of least privilege.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To limit container definitions to read-only access to root filesystems

1. Open the Amazon ECS console at <https://console.aws.amazon.com/ecs/>.
2. In the left navigation pane, choose **Task Definitions**.
3. For each task definition that has container definitions that need to be updated, do the following:
 - Select the container definition that needs to be updated.
 - Choose **Edit Container**. For **Storage and Logging**, select **Read only root file system**.
 - Choose **Update** at the bottom of the **Edit Container** tab.
 - Choose **Create**.

[ECS.8] Secrets should not be passed as container environment variables

Category: Protect > Secure development > Credentials not hard-coded

Severity: High

Resource type: AWS::ECS::TaskDefinition

AWS Configrule: [ecs-no-environment-secrets](#)

Schedule type: Change triggered

Parameters:

- secretKeys = AWS_ACCESS_KEY_ID,AWS_SECRET_ACCESS_KEY,ECS_ENGINE_AUTH_DATA

This control checks if the key value of any variables in the `environment` parameter of container definitions includes `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`. This control fails if a single environment variable in any container definition equals `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, or `ECS_ENGINE_AUTH_DATA`. This control does not cover environmental variables passed in from other locations such as Amazon S3.

AWS Systems Manager Parameter Store can help you improve the security posture of your organization. We recommend using the Parameter Store to store secrets and credentials instead of directing passing them into your container instances or hard coding them into your code.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To create parameters using SSM, see [Creating Systems Manager parameters](#) in the *AWS Systems Manager User Guide*. For more information about creating a task definition that specifies a secret, see [Specifying sensitive data using Secrets Manager](#) in the *Amazon Elastic Container Service Developer Guide*.

[ECS.10] Fargate services should run on the latest Fargate platform version

Category: Identify > Vulnerability, patch, and version management

Severity: Medium

Resource type: AWS::ECS::Service

AWS Configrule: [ecs-fargate-latest-platform-version](#)

Schedule type: Change triggered

Parameters: None

This control checks if Amazon ECS Fargate services are running the latest Fargate platform version. This control fails if the platform version is not the latest.

AWS Fargate platform versions refer to a specific runtime environment for Fargate task infrastructure, which is a combination of kernel and container runtime versions. New platform versions are released as the runtime environment evolves. For example, a new version may be released for kernel or operating system updates, new features, bug fixes, or security updates. Security updates and patches are deployed automatically for your Fargate tasks. If a security issue is found that affects a platform version, AWS patches the platform version.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To update an existing service, including its platform version, see [Updating a service](#) in the *Amazon Elastic Container Service Developer Guide*.

[ECS.12] ECS clusters should have Container Insights enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::ECS::Cluster

AWS Configrule: [ecs-container-insights-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks if ECS clusters use Container Insights. This control fails if Container Insights are not set up for a cluster.

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon ECS clusters. Use CloudWatch Container Insights to collect, aggregate, and summarize metrics and logs

from your containerized applications and microservices. CloudWatch automatically collects metrics for many resources, such as CPU, memory, disk, and network. Container Insights also provides diagnostic information, such as container restart failures, to help you isolate issues and resolve them quickly. You can also set CloudWatch alarms on metrics that Container Insights collects.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To use Container Insights, see [Updating a service](#) in the *Amazon CloudWatch User Guide*.

[EFS.1] Amazon EFS should be configured to encrypt file data at rest using AWS KMS

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::EFS::FileSystem

AWS Config rule: [efs-encrypted-check](#)

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic File System is configured to encrypt the file data using AWS KMS. The check fails in the following cases.

- `Encrypted` is set to `false` in the [DescribeFileSystems](#) response.
- The `KmsKeyId` key in the [DescribeFileSystems](#) response does not match the `KmsKeyId` parameter for [efs-encrypted-check](#).

Note that this control does not use the `KmsKeyId` parameter for [efs-encrypted-check](#). It only checks the value of `Encrypted`.

For an added layer of security for your sensitive data in Amazon EFS, you should create encrypted file systems. Amazon EFS supports encryption for file systems at-rest. You can enable encryption of data at rest when you create an Amazon EFS file system. To learn more about Amazon EFS encryption, see [Data encryption in Amazon EFS](#) in the *Amazon Elastic File System User Guide*.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

For details on how to encrypt a new Amazon EFS file system, see [Encrypting data at rest](#) in the *Amazon Elastic File System User Guide*.

[EFS.2] Amazon EFS volumes should be in backup plans

Category: Recover > Resilience > Backup

Severity: Medium

Resource type: AWS::EFS::FileSystem

AWS Config rule: [efs-in-backup-plan](#)

Schedule type: Periodic

Parameters: None

This control checks whether Amazon Elastic File System (Amazon EFS) file systems are added to the backup plans in AWS Backup. The control fails if Amazon EFS file systems are not included in the backup plans.

Including EFS file systems in the backup plans helps you to protect your data from deletion and data loss.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To remediate this issue, update your file system to enable automatic backups.

To enable automatic backups for an existing file system

1. Open the Amazon Elastic File System console at <https://console.aws.amazon.com/efs/>.
2. On the **File systems** page, choose the file system for which to enable automatic backups.

The **File system details** page is displayed.

3. Under **General**, choose **Edit**.
4. To enable automatic backups, select **Enable automatic backups**.
5. Choose **Save changes**.

To learn more, visit [Using AWS Backup with Amazon EFS](#) in the *Amazon Elastic File System User Guide*.

[EFS.3] EFS access points should enforce a root directory

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS::EFS::AccessPoint

AWS Config rule: [efs-access-point-enforce-root-directory](#)

Schedule type: Change triggered

Parameters: None

This control checks if Amazon EFS access points are configured to enforce a root directory. The control fails if the value of `Path` is set to `/` (the default root directory of the file system).

When you enforce a root directory, the NFS client using the access point uses the root directory configured on the access point instead of the file system's root directory. Enforcing a root directory for an access point helps restrict data access by ensuring that users of the access point can only reach files of the specified subdirectory.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For instructions on how to enforce a root directory for an Amazon EFS access point, see [Enforcing a root directory with an access point](#) in the *Amazon Elastic File System User Guide*.

[EFS.4] EFS access points should enforce a user identity

Category: Protect > Secure access management

Severity: Medium

Resource type: `AWS::EFS::AccessPoint`

AWS Config rule: `efs-access-point-enforce-user-identity`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon EFS access points are configured to enforce a user identity. This control fails if a POSIX user identity is not defined while creating the EFS access point.

Amazon EFS access points are application-specific entry points into an EFS file system that make it easier to manage application access to shared datasets. Access points can enforce a user identity, including the user's POSIX groups, for all file system requests that are made through the access point. Access points can also enforce a different root directory for the file system so that clients can only access data in the specified directory or its subdirectories.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To enforce a user identity for an Amazon EFS access point, see [Enforcing a user identity using an access point](#) in the *Amazon Elastic File System User Guide*.

[EKS.2] EKS clusters should run on a supported Kubernetes version

Category: Identify > Vulnerability, patch, and version management

Severity: High

Resource type: AWS::EKS::Cluster

AWS Config rule: [eks-cluster-supported-version](#)

Schedule type: Change triggered

Parameters:

- `eks:oldestVersionSupported` (Current oldest supported version is 1.19)

This control checks whether an Amazon EKS cluster is running on a supported Kubernetes version. The control fails if the EKS cluster is running on an unsupported version.

If your application doesn't require a specific version of Kubernetes, we recommend that you use the latest available Kubernetes version that's supported by EKS for your clusters. For more information about supported Kubernetes versions for Amazon EKS, see [Amazon EKS Kubernetes release calendar](#) and [Amazon EKS version support and FAQ/para>](#) in the Amazon EKS User Guide.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To update an EKS cluster, [Updating an Amazon EKS cluster Kubernetes version/para>](#) in the Amazon EKS User Guide.

[ElasticBeanstalk.1] Elastic Beanstalk environments should have enhanced health reporting enabled

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: AWS::ElasticBeanstalk::Environment

AWS Config rule: [beanstalk-enhanced-health-reporting-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether enhanced health reporting is enabled for your AWS Elastic Beanstalk environments.

Elastic Beanstalk enhanced health reporting enables a more rapid response to changes in the health of the underlying infrastructure. These changes could result in a lack of availability of the application.

Elastic Beanstalk enhanced health reporting provides a status descriptor to gauge the severity of the identified issues and identify possible causes to investigate. The Elastic Beanstalk health agent, included in supported Amazon Machine Images (AMIs), evaluates logs and metrics of environment EC2 instances.

For additional information, see [Enhanced health reporting and monitoring](#) in the *AWS Elastic Beanstalk Developer Guide*.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For instructions on how to enable enhanced health reporting, see [Enabling enhanced health reporting using the Elastic Beanstalk console](#) in the *AWS Elastic Beanstalk Developer Guide*.

[ElasticBeanstalk.2] Elastic Beanstalk managed platform updates should be enabled

Category: Detect > Vulnerability, patch, and version management

Severity: High

Resource type: AWS::ElasticBeanstalk::Environment

AWS Config rule: [elastic-beanstalk-managed-updates-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether managed platform updates are enabled for the Elastic Beanstalk environment.

Enabling managed platform updates ensures that the latest available platform fixes, updates, and features for the environment are installed. Keeping up to date with patch installation is an important step in securing systems.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For instructions on how to enable managed platform updates, see [To configure managed platform updates under Managed platform updates](#) in the *AWS Elastic Beanstalk Developer Guide*.

[ELB.2] Classic Load Balancers with HTTPS/SSL listeners should use a certificate provided by AWS Certificate Manager

Category: Protect > Encryption of data in transit

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [elb-acm-certificate-required](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the Classic Load Balancer uses HTTPS/SSL certificates provided by AWS Certificate Manager (ACM). The control fails if the Classic Load Balancer configured with HTTPS/SSL listener does not use a certificate provided by ACM.

To create a certificate, you can use either ACM or a tool that supports the SSL and TLS protocols, such as OpenSSL. Security Hub recommends that you use ACM to create or import certificates for your load balancer.

ACM integrates with Classic Load Balancers so that you can deploy the certificate on your load balancer. You also should automatically renew these certificates.

Note

These controls are not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- AWS GovCloud (US-East)

Remediation

For information on how to associate an ACM SSL/TLS certificate with a Classic Load Balancer, see the Knowledge Center article [How can I associate an ACM SSL/TLS certificate with a Classic, Application, or Network Load Balancer?](#)

[ELB.3] Classic Load Balancer listeners should be configured with HTTPS or TLS termination

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [elb-tls-https-listeners-only](#)

Schedule type: Change triggered

Parameters: None

This control checks whether your Classic Load Balancer listeners are configured with HTTPS or TLS protocol for front-end (client to load balancer) connections. The control is applicable if a Classic Load Balancer has listeners. If your Classic Load Balancer does not have a listener configured, then the control does not report any findings.

The control passes if the Classic Load Balancer listeners are configured with TLS or HTTPS for front-end connections.

The control fails if the listener is not configured with TLS or HTTPS for front-end connections.

Before you start to use a load balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners can support both HTTP and HTTPS/TLS protocols. You should always use an HTTPS or TLS listener, so that the load balancer does the work of encryption and decryption in transit.

Remediation

To remediate this issue, update your listeners to use the TLS or HTTPS protocol.

To change all noncompliant listeners to TLS/HTTPS listeners

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Load Balancers**. Then choose your Classic Load Balancer.
3. Choose the **Listeners** tab, and then choose **Edit**.
4. For all listeners where **Load Balancer Protocol** is not set to HTTPS or SSL, change the setting to HTTPS or SSL.
5. For all modified listeners, under **SSL Certificate**, choose **Change**.
6. For all modified listeners, select **Choose a certificate from ACM**.
7. Select the certificate from the **Certificates** drop-down list. Then choose **Save**.
8. After you update all of the listeners, choose **Save**.

[ELB.4] Application load balancers should be configured to drop HTTP headers

Category: Protect > Network security

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [alb-http-drop-invalid-header-enabled](#)

Schedule type: Change triggered

Parameters: None

This control evaluates AWS Application Load Balancers to ensure they are configured to drop invalid HTTP headers. The control fails if the value of `routing.http.drop_invalid_header_fields.enabled` is set to `false`.

By default, Application Load Balancers are not configured to drop invalid HTTP header values. Removing these header values prevents HTTP desync attacks.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)

- Asia Pacific (Osaka)
- Europe (Milan)

Remediation

To remediate this issue, configure your load balancer to drop invalid header fields.

To configure the load balancer to drop invalid header fields

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Load balancers**.
3. Choose an Application Load Balancer.
4. From **Actions**, choose **Edit attributes**.
5. Under **Drop Invalid Header Fields**, choose **Enable**.
6. Choose **Save**.

[ELB.5] Application and Classic Load Balancers logging should be enabled

Category: Logging

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: `elb-logging-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether the Application Load Balancer and the Classic Load Balancer have logging enabled. The control fails if `access_logs.s3.enabled` is `false`.

Elastic Load Balancing provides access logs that capture detailed information about requests sent to your load balancer. Each log contains information such as the time the request was received, the client's IP address, latencies, request paths, and server responses. You can use these access logs to analyze traffic patterns and to troubleshoot issues.

To learn more, see [Access logs for your Classic Load Balancer](#) in *User Guide for Classic Load Balancers*.

Remediation

To remediate this issue, update your load balancers to enable logging.

To enable access logs

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Load balancers**.
3. Choose an Application Load Balancer.
4. From **Actions**, choose **Edit attributes**.
5. Under **Access logs**, choose **Enable**.
6. Enter your S3 location. This location can exist or it can be created for you. If you do not specify a prefix, the access logs are stored in the root of the S3 bucket.
7. Choose **Save**.

[ELB.6] Application Load Balancer deletion protection should be enabled

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: AWS::ElasticLoadBalancingV2::LoadBalancer

AWS Config rule: [elb-deletion-protection-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Application Load Balancer has deletion protection enabled. The control fails if deletion protection is not configured.

Enable deletion protection to protect your Application Load Balancer from deletion.

Remediation

To prevent your load balancer from being deleted accidentally, you can enable deletion protection. By default, deletion protection is disabled for your load balancer.

If you enable deletion protection for your load balancer, you must disable delete protection before you can delete the load balancer.

To enable deletion protection from the console

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. On the navigation pane, under **LOAD BALANCING**, choose **Load Balancers**.
3. Choose the load balancer.
4. On the **Description** tab, choose **Edit attributes**.
5. On the **Edit load balancer attributes** page, select **Enable for Delete Protection**, and then choose **Save**.
6. Choose **Save**.

To learn more, see [Deletion protection](#) in *User Guide for Application Load Balancers*.

[ELB.7] Classic Load Balancers should have connection draining enabled

Category: Recover > Resilience

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [elb-connection-draining-enabled](#) (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether Classic Load Balancers have connection draining enabled.

Enabling connection draining on Classic Load Balancers ensures that the load balancer stops sending requests to instances that are de-registering or unhealthy. It keeps the existing connections open. This

is particularly useful for instances in Auto Scaling groups, to ensure that connections aren't severed abruptly.

Remediation

To enable connection draining on Classic Load Balancers, following the steps in [Configure connection draining for your Classic Load Balancer](#) in *User Guide for Classic Load Balancers*.

[ELB.8] Classic Load Balancers with HTTPS/SSL listeners should use a predefined security policy that has strong configuration

Category: Protect > Encryption of data in transit

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [elb-predefined-security-policy-ssl-check](#)

Schedule type: Change triggered

Parameters:

- `predefinedPolicyName: ELBSecurityPolicy-TLS-1-2-2017-01`

This control checks whether your Classic Load Balancer HTTPS/SSL listeners use the predefined policy `ELBSecurityPolicy-TLS-1-2-2017-01`. The control fails if the Classic Load Balancer HTTPS/SSL listeners do not use `ELBSecurityPolicy-TLS-1-2-2017-01`.

A security policy is a combination of SSL protocols, ciphers, and the Server Order Preference option. Predefined policies control the ciphers, protocols, and preference orders to support during SSL negotiations between a client and load balancer.

Using `ELBSecurityPolicy-TLS-1-2-2017-01` can help you to meet compliance and security standards that require you to disable specific versions of SSL and TLS. For more information, see [Predefined SSL security policies for Classic Load Balancers](#) in *User Guide for Classic Load Balancers*.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Europe (Milan)
- AWS GovCloud (US-East)

Remediation

For information on how to use the predefined security policy `ELBSecurityPolicy-TLS-1-2-2017-01` with a Classic Load Balancer, see [Configure security settings](#) in *User Guide for Classic Load Balancers*.

[ELB.9] Classic Load Balancers should have cross-zone load balancing enabled

Category: Recover > Resilience > High Availability

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: `elb-cross-zone-load-balancing-enabled`

Schedule type: Change triggered

Parameters:

- `predefinedPolicyName: ELBSecurityPolicy-TLS-1-2-2017-01`

This control checks if cross-zone load balancing is enabled for the Classic Load Balancers (CLBs). This control fails if cross-zone load balancing is not enabled for a CLB.

A load balancer node distributes traffic only across the registered targets in its Availability Zone. When cross-zone load balancing is disabled, each load balancer node distributes traffic only across the registered targets in its Availability Zone. If the number of registered targets is not same across the Availability Zones, traffic wont be distributed evenly and the instances in one zone may end up over utilized compared to the instances in another zone. With cross-zone load balancing enabled, each load balancer node for your Classic Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. For details see [Cross-zone load balancing](#) in the Elastic Load Balancing User Guide.

Note

This control is not supported in Africa (Cape Town).

Remediation

To enable cross-zone load balancing in a Classic Load Balancer, see [Enable cross-zone load balancing](#) in the Elastic Load Balancing User Guide.

[ELB.10] Classic Load Balancers should span multiple Availability Zones

Category: Recover > Resilience > High Availability

Severity: Medium

Resource type: `AWS::ElasticLoadBalancing::LoadBalancer`

AWS Config rule: `clb-multiple-az`

Schedule type: Change triggered

Parameters: None

This control checks whether a Classic Load Balancer has been configured to span multiple Availability Zones. The control fails if the Classic Load Balancer does not span multiple Availability Zones.

A Classic Load Balancer can be set up to distribute incoming requests across Amazon EC2 instances in a single Availability Zone or multiple Availability Zones. A Classic Load Balancer that does not span multiple Availability Zones is unable to redirect traffic to targets in another Availability Zone if the sole configured Availability Zone becomes unavailable.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information on how to add Availability Zones to a Classic Load Balancer, see [Add or remove Availability Zones](#) in the *User Guide for Classic Load Balancers*.

[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode

Category: Data protect > Data integrity

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [alb-desync-mode-check](#)

Schedule type: Change triggered

Parameters:

- `desyncMode`: defensive, strictest

This control checks whether an Application Load Balancer is configured with defensive or strictest desync mitigation mode. The control fails if an Application Load Balancer is not configured with defensive or strictest desync mitigation mode.

HTTP Desync issues can lead to request smuggling and make applications vulnerable to request queue or cache poisoning. In turn, these vulnerabilities can lead to credential hijacking or execution of unauthorized commands. Application Load Balancers configured with defensive or strictest desync mitigation mode protect your application from security issues that may be caused by HTTP Desync.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To update desync mitigation mode of an Application Load Balancer, see [Desync mitigation mode](#) in the *User Guide for Application Load Balancers*.

[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: AWS::ElasticLoadBalancingV2::LoadBalancer

AWS Config rule: [elbv2-multiple-az](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Elastic Load Balancer V2 (Application, Network, or Gateway Load Balancer) has registered instances from multiple Availability Zones. The control fails if an Elastic Load Balancer V2 has instances registered in fewer than two Availability Zones.

Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones. Elastic Load Balancing scales your load balancer as your incoming traffic changes over time. It is recommended to configure at least two availability zones to ensure availability of services, as the Elastic Load Balancer will be able to direct traffic to another availability zone if one becomes unavailable. Having multiple availability zones configured will help eliminate having a single point of failure for the application.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To add an Availability Zone to an Application Load Balancer, see [Availability Zones for your Application Load Balancer](#) in the *User Guide for Application Load Balancers*. To add an Availability Zone to a Network Load Balancer, see [Network Load Balancers](#) in the *User Guide for Network Load Balancers*. To add an Availability Zone to a Gateway Load Balancer, see [Create a Gateway Load Balancer](#) in the *User Guide for Gateway Load Balancers*.

[ELB.14] Classic Load Balancers should be configured with defensive or strictest desync mitigation mode

Category: Data Protect > Data Integrity

Severity: Medium

Resource type: AWS::ElasticLoadBalancing::LoadBalancer

AWS Config rule: [clb-desync-mode-check](#)

Schedule type: Change triggered

Parameters:

- `desyncMode`: defensive, strictest

This control checks whether a Classic Load Balancer is configured with defensive or strictest desync mitigation mode. This control will fail if the Application Load Balancer is not configured with defensive or strictest desync mitigation mode.

HTTP Desync issues can lead to request smuggling and make applications vulnerable to request queue or cache poisoning. In turn, these vulnerabilities can lead to credential hijacking or execution of unauthorized commands. Classic Load Balancers configured with defensive or strictest desync mitigation mode protect your application from security issues that may be caused by HTTP Desync.

Note

This control is not supported in the following Regions:

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To update desync mitigation mode of a Classic Load Balancer, see [Modify desync mitigation mode](#) in the *User Guide for Classic Load Balancers*.

[ELBv2.1] Application Load Balancer should be configured to redirect all HTTP requests to HTTPS

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::ElasticLoadBalancingV2::LoadBalancer

AWS Config rule: [alb-http-to-https-redirection-check](#)

Schedule type: Periodic

Parameters: None

This control checks whether HTTP to HTTPS redirection is configured on all HTTP listeners of Application Load Balancers. The check fails if one or more HTTP listeners of Application Load Balancers do not have HTTP to HTTPS redirection configured.

Before you start to use your Application Load Balancer, you must add one or more listeners. A listener is a process that uses the configured protocol and port to check for connection requests. Listeners support both HTTP and HTTPS protocols. You can use an HTTPS listener to offload the work of encryption and decryption to your Application Load Balancer. You should use redirect actions with Application Load Balancer to redirect client HTTP request to an HTTPS request on port 443 to enforce encryption in-transit.

To learn more, see [Listeners for your Application Load Balancers](#) in *User Guide for Application Load Balancers*.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

To remediate this issue, update your load balancers to redirect HTTP requests.

To redirect HTTP requests to HTTPS on an Application Load Balancer

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. In the navigation pane, choose **Load balancers**.
3. Choose an Application Load Balancer.
4. Choose the **Listeners** tab.
5. Choose an HTTP listener (port 80 TCP) and then choose **Edit**.
6. If there is an existing rule, you must delete it. Otherwise, choose **Add action** and then choose **Redirect to...**
7. Choose **HTTPS** and then enter **443**.
8. Choose the check mark in a circle symbol and then choose **Update**.

[EMR.1] Amazon EMR cluster master nodes should not have public IP addresses

Category: Protect > Secure network configuration

Severity: High

Resource type: AWS::EMR::Cluster

AWS Config rule: [emr-master-no-public-ip](#)

Schedule type: Periodic

Parameters: None

This control checks whether master nodes on Amazon EMR clusters have public IP addresses.

The control fails if the master node has public IP addresses that are associated with any of its instances. Public IP addresses are designated in the `PublicIp` field of the `NetworkInterfaces` configuration for the instance. This control only checks Amazon EMR clusters that are in a `RUNNING` or `WAITING` state.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

During launch, you can control whether your instance in a default or nondefault subnet is assigned a public IPv4 address.

By default, default subnets have this attribute set to `true`. Nondefault subnets have the IPv4 public addressing attribute set to `false`, unless it was created by the Amazon EC2 launch instance wizard. In that case, the wizard sets the attribute to `true`.

You need to launch your cluster in a VPC with a private subnet that has the IPv4 public addressing attribute set to `false`.

After launch, you cannot manually disassociate a public IPv4 address from your instance.

To remediate this finding, you need to create a new cluster in VPC private subnet. For information on how to launch a cluster in into a VPC private subnet, see [Launch clusters into a VPC](#) in the *Amazon EMR Management Guide*.

[ES.1] Elasticsearch domains should have encryption at rest enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: [elasticsearch-encrypted-at-rest](#)

Schedule type: Periodic

Parameters: None

This control checks whether Elasticsearch domains have encryption at rest configuration enabled. The check fails if encryption at rest is not enabled.

For an added layer of security for your sensitive data in OpenSearch, you should configure your OpenSearch to be encrypted at rest. Elasticsearch domains offer encryption of data at rest. The feature

uses AWS KMS to store and manage your encryption keys. To perform the encryption, it uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

To learn more about OpenSearch encryption at rest, see [Encryption of data at rest for Amazon OpenSearch Service](#) in the *Amazon OpenSearch Service Developer Guide*.

Note

Certain instance types, such as `t.small` and `t.medium`, do not support encryption of data at rest. For details, see [Supported instance types](#) in the *Amazon OpenSearch Service Developer Guide*.

Remediation

By default, domains do not encrypt data at rest, and you cannot configure existing domains to use the feature.

To enable the feature, you must create another domain and migrate your data. For information about creating domains, see the *Amazon OpenSearch Service Developer Guide*.

Encryption of data at rest requires OpenSearch Service 5.1 or later. For more information about encrypting data at rest for OpenSearch Service, see the *Amazon OpenSearch Service Developer Guide*.

[ES.2] Elasticsearch domains should be in a VPC

Category: Protect > Secure network configuration > Resources within VPC

Severity: Critical

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: [elasticsearch-in-vpc-only](#)

Schedule type: Periodic

Parameters: None

This control checks whether Elasticsearch domains are in a VPC. It does not evaluate the VPC subnet routing configuration to determine public access. You should ensure that Elasticsearch domains are not attached to public subnets. See [Resource-based policies](#) in the *Amazon OpenSearch Service Developer Guide*. You should also ensure that your VPC is configured according to the recommended best practices. See [Security best practices for your VPC](#) in the *Amazon VPC User Guide*.

Elasticsearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls to secure access to Elasticsearch domains, including network ACL and security groups. Security Hub recommends that you migrate public Elasticsearch domains to VPCs to take advantage of these controls.

Remediation

If you create a domain with a public endpoint, you cannot later place it within a VPC. Instead, you must create a new domain and migrate your data. The reverse is also true. If you create a domain within a VPC, it cannot have a public endpoint. Instead, you must either [create another domain](#) or disable this control.

See [Launching your Amazon OpenSearch Service domains within a VPC](#) in the *Amazon OpenSearch Service Developer Guide*.

[ES.3] Elasticsearch domains should encrypt data sent between nodes

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: [elasticsearch-node-to-node-encryption-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Elasticsearch domains have node-to-node encryption enabled.

HTTPS (TLS) can be used to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks. Only encrypted connections over HTTPS (TLS) should be allowed. Enabling node-to-node encryption for Elasticsearch domains ensures that intra-cluster communications are encrypted in transit.

There can be a performance penalty associated with this configuration. You should be aware of and test the performance trade-off before enabling this option.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)

Remediation

For information about enabling node-to-node encryption on new and existing domains, see [Enabling node-to-node encryption](#) in the *Amazon OpenSearch Service Developer Guide*.

[ES.4] Elasticsearch domain error logging to CloudWatch Logs should be enabled

Category: Identify - Logging

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: [elasticsearch-logs-to-cloudwatch](#)

Schedule type: Change triggered

Parameters:

- `logtype = 'error'`

This control checks whether Elasticsearch domains are configured to send error logs to CloudWatch Logs.

You should enable error logs for Elasticsearch domains and send those logs to CloudWatch Logs for retention and response. Domain error logs can assist with security and access audits, and can help to diagnose availability issues.

Note

This control is not supported in the following Regions:

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information on how to enable log publishing, see [Enabling log publishing \(console\)](#) in the *Amazon OpenSearch Service Developer Guide*.

[ES.5] Elasticsearch domains should have audit logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: `elasticsearch-audit-logging-enabled` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters:

- `cloudWatchLogsLogGroupArnList` (Optional). Security Hub does not populate this parameter. Comma-separated list of CloudWatch Logs log groups that should be configured for audit logs.

This rule is `NON_COMPLIANT` if the CloudWatch Logs log group of the Elasticsearch domain is not specified in this parameter list.

This control checks whether Elasticsearch domains have audit logging enabled. This control fails if an Elasticsearch domain does not have audit logging enabled.

Audit logs are highly customizable. They allow you to track user activity on your Elasticsearch clusters, including authentication successes and failures, requests to OpenSearch, index changes, and incoming search queries.

Remediation

For detailed instructions on enabling audit logs, see [Enabling audit logs](#) in the *Amazon OpenSearch Service Developer Guide*.

[ES.6] Elasticsearch domains should have at least three data nodes

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: `elasticsearch-data-node-fault-tolerance` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether Elasticsearch domains are configured with at least three data nodes and `zoneAwarenessEnabled` is true.

An Elasticsearch domain requires at least three data nodes for high availability and fault-tolerance. Deploying an Elasticsearch domain with at least three data nodes ensures cluster operations if a node fails.

Remediation

To modify the number of data nodes in an Elasticsearch domain

1. Open the Amazon OpenSearch Service console at <https://console.aws.amazon.com/es/>.
2. Under **My domains**, choose the name of the domain to edit.
3. Choose **Edit domain**.
4. Under **Data nodes**, set **Number of nodes** to a number greater than or equal to 3.

For three Availability Zone deployments, set to a multiple of three to ensure equal distribution across Availability Zones.

5. Choose **Submit**.

[ES.7] Elasticsearch domains should be configured with at least three dedicated master nodes

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Configrule: `elasticsearch-primary-node-fault-tolerance` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether Elasticsearch domains are configured with at least three dedicated master nodes. This control fails if the domain does not use dedicated master nodes. This control passes if Elasticsearch domains have five dedicated master nodes. However, using more than three master nodes might be unnecessary to mitigate the availability risk, and will result in additional cost.

An Elasticsearch domain requires at least three dedicated master nodes for high availability and fault-tolerance. Dedicated master node resources can be strained during data node blue/green deployments because there are additional nodes to manage. Deploying an Elasticsearch domain with at least three dedicated master nodes ensures sufficient master node resource capacity and cluster operations if a node fails.

Remediation

To modify the number of dedicated master nodes in an OpenSearch domain

1. Open the Amazon OpenSearch Service console at <https://console.aws.amazon.com/es/>.
2. Under **My domains**, choose the name of the domain to edit.
3. Choose **Edit domain**.
4. Under **Dedicated master nodes**, set **Instance type** to the desired instance type.
5. Set **Number of master nodes** equal to three or greater.

6. Choose **Submit**.

[ES.8] Connections to Elasticsearch domains should be encrypted using TLS 1.2

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::Elasticsearch::Domain

AWS Config rule: `elasticsearch-https-required` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether connections to Elasticsearch domains are required to use TLS 1.2. The check fails if the Elasticsearch domain `TLSecurityPolicy` is not `Policy-Min-TLS-1-2-2019-07`.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS. TLS 1.2 provides several security enhancements over previous versions of TLS.

Remediation

To enable TLS encryption, use the [UpdateDomainConfig](#) API operation to configure the `DomainEndpointOptions` in order to set the `TLSecurityPolicy`. For more information, see the *Amazon OpenSearch Service Developer Guide*.

[GuardDuty.1] GuardDuty should be enabled

Category: Detect > Detection services

Severity: High

Resource type: AWS account

AWS Config rule: `guardduty-enabled-centralized`

Schedule type: Periodic

Parameters: None

This control checks whether Amazon GuardDuty is enabled in your GuardDuty account and Region.

It is highly recommended that you enable GuardDuty in all supported AWS Regions. Doing so allows GuardDuty to generate findings about unauthorized or unusual activity, even in Regions that you do not actively use. This also allows GuardDuty to monitor CloudTrail events for global AWS services such as IAM.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)

- Middle East (Bahrain)
- AWS GovCloud (US-East)

Remediation

To remediate this issue, you enable GuardDuty.

For details on how to enable GuardDuty, including how to use AWS Organizations to manage multiple accounts, see [Getting started with GuardDuty](#) in the *Amazon GuardDuty User Guide*.

[IAM.1] IAM policies should not allow full "*" administrative privileges

Category: Protect > Secure access management

Severity: High

Resource type: AWS::IAM::Policy

AWS Config rule: [iam-policy-no-statements-with-admin-access](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the default version of IAM policies (also known as customer managed policies) has administrator access that includes a statement with "Effect": "Allow" with "Action": "*" over "Resource": "*".

The control only checks the customer managed policies that you create. It does not check inline and AWS managed policies.

IAM policies define a set of privileges that are granted to users, groups, or roles. Following standard security advice, AWS recommends that you grant least privilege, which means to grant only the permissions that are required to perform a task. When you provide full administrative privileges instead of the minimum set of permissions that the user needs, you expose the resources to potentially unwanted actions.

Instead of allowing full administrative privileges, determine what users need to do and then craft policies that let the users perform only those tasks. It is more secure to start with a minimum set of permissions and grant additional permissions as necessary. Do not start with permissions that are too lenient and then try to tighten them later.

You should remove IAM policies that have a statement with "Effect": "Allow" with "Action": "*" over "Resource": "*".

Remediation

To modify your IAM policies so that they do not allow full "*" administrative privileges, see [Editing IAM policies](#) in the *IAM User Guide*.

[IAM.2] IAM users should not have IAM policies attached

Category: Protect > Secure access management

Severity: Low

Resource type: AWS::IAM::User

AWS Config rule: [iam-user-no-policies-check](#)

Schedule type: Change triggered

Parameters: None

This control checks that none of your IAM users have policies attached. Instead, IAM users must inherit permissions from IAM groups or roles.

By default, IAM users, groups, and roles have no access to AWS resources. IAM policies grant privileges to users, groups, or roles. We recommend that you apply IAM policies directly to groups and roles but not to users. Assigning privileges at the group or role level reduces the complexity of access management as the number of users grows. Reducing access management complexity might in turn reduce the opportunity for a principal to inadvertently receive or retain excessive privileges.

Note

IAM users created by Amazon Simple Email Service are automatically created using inline policies. Security Hub automatically exempts these users from this control.

Remediation

To resolve this issue, [create an IAM group](#), and attach the policy to the group. Then, [add the users to the group](#). The policy is applied to each user in the group. To remove a policy attached directly to a user, see [Deleting IAM policies](#) in the *IAM User Guide*.

[IAM.3] IAM users' access keys should be rotated every 90 days or less

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS : IAM : User

AWS Config rule: [access-keys-rotated](#)

Schedule type: Periodic

Parameters:

- `maxAccessKeyAge: 90`

This control checks whether the active access keys are rotated within 90 days.

We highly recommend that you do not generate and remove all access keys in your account. Instead, the recommended best practice is to either create one or more IAM roles or to use [federation](#). You can use these methods to allow your users to use their existing corporate credentials to log into the AWS Management Console and AWS CLI.

Each approach has its use cases. Federation is generally better for enterprises that have an existing central directory or plan to need more than the current limit IAM users. Applications that run outside of an AWS environment need access keys for programmatic access to AWS resources.

However, if the resources that need programmatic access run inside AWS, the best practice is to use IAM roles. Roles allow you to grant a resource access without hardcoding an access key ID and secret access key into the configuration.

To learn more about protecting your access keys and account, see [Best practices for managing AWS access keys](#) in the *AWS General Reference*. Also see the blog post [Guidelines for protecting your AWS account while using programmatic access](#).

If you already have an access key, Security Hub recommends that you rotate the access keys every 90 days. Rotating access keys reduces the chance that an access key that is associated with a compromised

or terminated account is used. It also ensures that data cannot be accessed with an old key that might have been lost, cracked, or stolen. Always update your applications after you rotate access keys.

Access keys consist of an access key ID and a secret access key. They are used to sign programmatic requests that you make to AWS. AWS users need their own access keys to make programmatic calls to AWS from the AWS CLI, Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the API operations for individual AWS services.

If your organization uses AWS Single Sign-On (AWS SSO), your users can sign in to Active Directory, a built-in AWS SSO directory, or [another identity provider \(IdP\) connected to AWS SSO](#). They can then be mapped to an IAM role that enables them to run AWS CLI commands or call AWS API operations without the need for IAM user access keys. To learn more, see [Configuring the AWS CLI to use AWS Single Sign-On](#) in the *AWS Command Line Interface User Guide*.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

To remediate this issue, replace any keys that are older than 90 days.

To ensure that access keys aren't more than 90 days old

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. For each user that shows an **Access key age** that is greater than 90 days, choose the **User name** to open the settings for that user.
4. Choose **Security credentials**.
5. Create a new key for the user:
 - a. Choose **Create access key**.
 - b. To save the key content, either download the secret access key, or choose **Show** and then copy it from the page.
 - c. Store the key in a secure location to provide to the user.
 - d. Choose **Close**.
6. Update all applications that were using the previous key to use the new key.
7. For the previous key, choose **Make inactive** to make the access key inactive. The user now cannot use that key to make requests.
8. Confirm that all applications work as expected with the new key.
9. After confirming that all applications work with the new key, delete the previous key. After you delete the access key, you cannot recover it.

To delete the previous key, choose the **X** at the end of the row and then choose **Delete**.

[IAM.4] IAM root user access key should not exist

Category: Protect > Secure access management

Severity: Critical

Resource type: AWS account

AWS Config rule: [iam-root-access-key-check](#)

Schedule type: Periodic

Parameters: None

This control checks whether the root user access key is present.

The root user is the most privileged user in an AWS account. AWS access keys provide programmatic access to a given account.

Security Hub recommends that you remove all access keys that are associated with the root user. This limits the vectors that can be used to compromise your account. It also encourages the creation and use of role-based accounts that are least privileged.

Note

This control is not supported in Africa (Cape Town).

Remediation

To remediate this issue, delete the root user access key.

To delete access keys

1. Log in to your account using the AWS account root user credentials.
2. Choose the account name near the top-right corner of the page and then choose **Security Credentials**.
3. Choose **Access keys (access key ID and secret access key)**.
4. To permanently delete the key, choose **Delete** and then choose **Yes**. You cannot recover deleted keys.
5. If there is more than one root user access key, then repeat steps 4 and 5 for each key.

[IAM.5] MFA should be enabled for all IAM users that have a console password

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS::IAM::User

AWS Config rule: [mfa-enabled-for-iam-console-access](#)

Schedule type: Periodic

Parameters: None

This control checks whether AWS multi-factor authentication (MFA) is enabled for all IAM users that use a console password.

Multi-factor authentication (MFA) adds an extra layer of protection on top of a user name and password. With MFA enabled, when a user signs in to an AWS website, they are prompted for their user name and password. In addition, they are prompted for an authentication code from their AWS MFA device.

We recommend that you enable MFA for all accounts that have a console password. MFA is designed to provide increased security for console access. The authenticating principal must possess a device that emits a time-sensitive key and must have knowledge of a credential.

Remediation

To remediate this issue, add MFA to users that do not yet have it.

To configure MFA for a user

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.

2. Choose **Users**.
3. Choose the **User name** of the user to configure MFA for.
4. Choose **Security credentials**.
5. Next to **Assigned MFA Device**, choose **Manage**.
6. Follow the **Manage MFA Device** wizard to assign the type of device appropriate for your environment.

To learn how to delegate MFA setup to users, see the blog post [How to delegate management of multi-factor authentication to AWS IAM users](#).

[IAM.6] Hardware MFA should be enabled for the root user

Category: Protect > Secure access management

Severity: Critical

Resource type: AWS account

AWS Config rule: `root-account-hardware-mfa-enabled`

Schedule type: Periodic

Parameters: None

This control checks whether your AWS account is enabled to use a hardware multi-factor authentication (MFA) device to sign in with root user credentials.

Virtual MFA might not provide the same level of security as hardware MFA devices. We recommend that you use only a virtual MFA device while you wait for hardware purchase approval or for your hardware to arrive. To learn more, see [Enabling a virtual multi-factor authentication \(MFA\) device \(console\)](#) in the *IAM User Guide*.

Both time-based one-time password (TOTP) and Universal 2nd Factor (U2F) tokens are viable as hardware MFA options.

Note

This control is not supported in the following Regions:

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West).

Remediation

To remediate this issue, add hardware-based MFA to the root user.

To enable hardware-based MFA for the root user

1. Log in to your account using the root user credentials.
2. Choose the account name near the top-right corner of the page and then choose **My Security Credentials**.
3. In the pop-up warning, choose **Continue to Security Credentials**.
4. Choose **Multi-Factor Authentication (MFA)**.
5. Choose **Activate MFA**.

6. Choose a hardware-based (not virtual) device to use for MFA and then choose **Continue**.
7. Complete the steps to configure the device type appropriate to your selection.

[IAM.7] Password policies for IAM users should have strong configurations

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS account

AWS Config rule: [iam-password-policy](#)

Schedule type: Periodic

Parameters:

- `RequireUppercaseCharacters: true`
- `RequireLowercaseCharacters: true`
- `RequireSymbols: true`
- `RequireNumbers: true`
- `MinimumPasswordLength: 8`

This control checks whether the account password policy for IAM users uses the following recommended configurations.

- `RequireUppercaseCharacters: true`
- `RequireLowercaseCharacters: true`
- `RequireSymbols: true`
- `RequireNumbers: true`
- `MinimumPasswordLength: 8`

To access the AWS Management Console, IAM users need passwords. As a best practice, Security Hub highly recommends that instead of creating IAM users, you use federation. Federation allows users to use their existing corporate credentials to log into the AWS Management Console. Use AWS Single Sign-On (AWS SSO) to create or federate the user, and then assume an IAM role into an account.

To learn more about identity providers and federation, see [Identity providers and federation](#) in the *IAM User Guide*. To learn more about AWS SSO, see the [AWS Single Sign-On User Guide](#).

If you need to use IAM users, Security Hub recommends that you enforce the creation of strong user passwords. You can set a password policy on your AWS account to specify complexity requirements and mandatory rotation periods for passwords. When you create or change a password policy, most of the password policy settings are enforced the next time users change their passwords. Some of the settings are enforced immediately. To learn more, see [Setting an account password policy for IAM users](#) in the *IAM User Guide*.

Remediation

To remediate this issue, update your password policy to use the recommended configuration.

To modify the password policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Account settings**.

3. Select **Requires at least one uppercase letter**.
4. Select **Requires at least one lowercase letter**.
5. Select **Requires at least one non-alphanumeric character**.
6. Select **Requires at least one number**.
7. For **Minimum password length**, enter **8**.
8. Choose **Apply password policy**.

[IAM.8] Unused IAM user credentials should be removed

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS :: IAM :: User

AWS Config rule: [iam-user-unused-credentials-check](#)

Schedule type: Periodic

Parameters:

- `maxCredentialUsageAge`: 90

This control checks whether your IAM users have passwords or active access keys that have not been used for 90 days.

IAM users can access AWS resources using different types of credentials, such as passwords or access keys.

Security Hub recommends that you remove or deactivate all credentials that were unused for 90 days or more. Disabling or removing unnecessary credentials reduces the window of opportunity for credentials associated with a compromised or abandoned account to be used.

Remediation

To get some of the information that you need to monitor accounts for dated credentials, use the IAM console. For example, when you view users in your account, there is a column for **Access key age**, **Password age**, and **Last activity**. If the value in any of these columns is greater than 90 days, make the credentials for those users inactive.

You can also use credential reports to monitor user accounts and identify those with no activity for 90 or more days. You can download credential reports in `.csv` format from the IAM console. For more information about credential reports, see [Getting credential reports for your AWS account](#) in the *IAM User Guide*.

After you identify the inactive accounts or unused credentials, use the following steps to disable them.

To disable credentials for inactive accounts

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. Choose **Users**.
3. Choose the name of the user that has credentials over 90 days old.
4. Choose **Security credentials**.
5. For each sign-in credential and access key that hasn't been used in at least 90 days, choose **Make inactive**.

[IAM.21] IAM customer managed policies that you create should not allow wildcard actions for services

Category: Detect > Secure access management

Severity: Low

Resource type: AWS::IAM::Policy

AWS Config rule: [iam-policy-no-statements-with-full-access](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the IAM identity-based policies that you create have Allow statements that use the * wildcard to grant permissions for all actions on any service. The control fails if any policy statement includes "Effect": "Allow" with "Action": "Service:*".

For example, the following statement in a policy results in a failed finding.

```
"Statement": [  
  {  
    "Sid": "EC2-Wildcard",  
    "Effect": "Allow",  
    "Action": "ec2:*",  
    "Resource": "*" } ]
```

The control also fails if you use "Effect": "Allow" with "NotAction": "*service**". In that case, the NotAction element provides access to all of the actions in an AWS service, except for the actions specified in NotAction.

This control only applies to customer managed IAM policies. It does not apply to IAM policies that are managed by AWS.

When you assign permissions to AWS services, it is important to scope the allowed IAM actions in your IAM policies. You should restrict IAM actions to only those actions that are needed. This helps you to provision least privilege permissions. Overly permissive policies might lead to privilege escalation if the policies are attached to an IAM principal that might not require the permission.

In some cases, you might want to allow IAM actions that have a similar prefix, such as DescribeFlowLogs and DescribeAvailabilityZones. In these authorized cases, you can add a suffixed wildcard to the common prefix. For example, ec2:Describe*.

This control passes if you use a prefixed IAM action with a suffixed wildcard. For example, the following statement in a policy results in a passed finding.

```
"Statement": [  
  {  
    "Sid": "EC2-Wildcard",  
    "Effect": "Allow",  
    "Action": "ec2:Describe*",  
    "Resource": "*" } ]
```

When you group related IAM actions in this way, you can also avoid exceeding the IAM policy size limits.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To remediate this issue, update your IAM policies so that they do not allow full "*" administrative privileges. For details about how to edit an IAM policy, see [Editing IAM policies](#) in the *IAM User Guide*.

[Kinesis.1] Kinesis Data Streams should be encrypted at rest

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::Kinesis::Stream

AWS Config rule: [kinesis-stream-encrypted](#)

Schedule type: Change triggered

Parameters: None

This control checks if Kinesis Data Streams are encrypted at rest with server-side encryption. This control fails if a Kinesis stream is not encrypted at rest with server-side encryption.

Server-side encryption is a feature in Amazon Kinesis Data Streams that automatically encrypts data before it's at rest by using an AWS KMS key. Data is encrypted before it's written to the Kinesis stream storage layer, and decrypted after it's retrieved from storage. As a result, your data is encrypted at rest within the Amazon Kinesis Data Streams service.

Note

This control is not supported in the following Regions:

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information about enabling server-side encryption for Kinesis streams, see [How Do I Get Started with Server-Side Encryption?](#) in the *Amazon Kinesis Developer Guide*.

[KMS.1] IAM customer managed policies should not allow decryption and re-encryption actions on all KMS keys

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS::IAM::Policy

AWS Config rule: [iam-customer-policy-blocked-kms-actions](#)

Schedule type: Change triggered

Parameters:

- `blockedActionsPatterns`: `kms:ReEncryptFrom`, `kms:Decrypt`

Checks whether the default version of IAM customer managed policies allow principals to use the AWS KMS decryption actions on all resources. This control uses [Zelkova](#), an automated reasoning engine, to validate and warn you about policies that may grant broad access to your secrets across AWS accounts.

This control fails, and flags the policy as `FAILED`, if the policy is open enough to allow `kms:Decrypt` or `kms:ReEncryptFrom` actions on any arbitrary KMS key.

The control evaluates both attached and unattached customer managed policies. It does not check inline policies or AWS managed policies.

With AWS KMS, you control who can use your KMS keys and gain access to your encrypted data. IAM policies define which actions an identity (user, group, or role) can perform on which resources. Following security best practices, AWS recommends that you allow least privilege. In other words, you should grant to identities only the `kms:Decrypt` or `kms:ReEncryptFrom` permissions and only for the keys that are required to perform a task. Otherwise, the user might use keys that are not appropriate for your data.

Instead of granting permissions for all keys, determine the minimum set of keys that users need to access encrypted data. Then design policies that allow users to use only those keys. For example, do not allow `kms:Decrypt` permission on all KMS keys. Instead, allow `kms:Decrypt` only on keys in a particular Region for your account. By adopting the principle of least privilege, you can reduce the risk of unintended disclosure of your data.

Remediation

To remediate this issue, you modify the IAM customer managed policies to restrict access to the keys.

To modify an IAM customer managed policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the IAM navigation pane, choose **Policies**.
3. Choose the arrow next to the policy you want to modify.
4. Choose **Edit policy**.
5. Choose the **JSON** tab.
6. Change the "Resource" value to the specific key or keys that you want to allow.
7. After you modify the policy, choose **Review policy**.
8. Choose **Save changes**.

For more information, see [Using IAM policies with AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

[KMS.2] IAM principals should not have IAM inline policies that allow decryption and re-encryption actions on all KMS keys

Category: Protect > Secure access management

Severity: Medium

Resource type:

- `AWS::IAM::Role`
- `AWS::IAM::User`
- `AWS::IAM::Group`

AWS Config rule: [iam-inline-policy-blocked-kms-actions](#)

Schedule type: Change triggered

Parameters:

- `blockedActionsPatterns`: `kms:ReEncryptFrom`, `kms:Decrypt`

Checks whether the inline policies that are embedded in your IAM identities (role, user, or group) allow the AWS KMS decryption and re-encryption actions on all KMS keys. This control uses [Zelkova](#), an automated reasoning engine, to validate and warn you about policies that may grant broad access to your secrets across AWS accounts.

This control fails if the policy is open enough to allow `kms:Decrypt` or `kms:ReEncryptFrom` actions on any arbitrary KMS key.

With AWS KMS, you control who can use your KMS keys and gain access to your encrypted data. IAM policies define which actions an identity (user, group, or role) can perform on which resources. Following security best practices, AWS recommends that you allow least privilege. In other words, you should grant to identities only the permissions they need and only for keys that are required to perform a task. Otherwise, the user might use keys that are not appropriate for your data.

Instead of granting permission for all keys, determine the minimum set of keys that users need to access encrypted data. Then design policies that allow the users to use only those keys. For example, do not allow `kms:Decrypt` permission on all KMS keys. Instead, allow the permission only on specific keys in a specific Region for your account. By adopting the principle of least privilege, you can reduce the risk of unintended disclosure of your data.

Remediation

To remediate this issue, you modify the inline policy to restrict access to the keys.

To modify an IAM inline policy

1. Open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the IAM navigation pane, choose **Users**, **Groups**, or **Roles**.
3. Choose the name of the user, group or role for which to modify IAM inline policies.
4. Choose the arrow next to the policy to modify.
5. Choose **Edit policy**.
6. Choose the **JSON** tab.
7. Change the "Resource" value to the specific keys you want to allow.
8. After you modify the policy, choose **Review policy**.
9. Choose **Save changes**.

For more information, see [Using IAM policies with AWS KMS](#) in the *AWS Key Management Service Developer Guide*.

[KMS.3] AWS KMS keys should not be unintentionally deleted

Category: Protect > Data protection > Data deletion protection

Severity: Critical

Resource type: AWS::KMS::Key

AWS Config rule: [kms-cmk-not-scheduled-for-deletion](#)

Schedule type: Periodic

Parameters: None

This control checks whether KMS keys are scheduled for deletion. The control fails if a KMS key is scheduled for deletion.

KMS keys cannot be recovered once deleted. Data encrypted under a KMS key is also permanently unrecoverable if the KMS key is deleted. If meaningful data has been encrypted under a KMS key scheduled for deletion, consider decrypting the data or re-encrypting the data under a new KMS key unless you are intentionally performing a *cryptographic erasure*.

When a KMS key is scheduled for deletion, a mandatory waiting period is enforced to allow time to reverse the deletion, if it was scheduled in error. The default waiting period is 30 days, but it can be reduced to as short as 7 days when the KMS key is scheduled for deletion. During the waiting period, the scheduled deletion can be canceled and the KMS key will not be deleted.

For additional information regarding deleting KMS keys, see [Deleting KMS keys](#) in the *AWS Key Management Service Developer Guide*.

Note

This control is not supported in the Asia Pacific (Osaka) and Europe (Milan) Regions.

Remediation

For detailed remediation instructions to cancel a scheduled KMS key deletion, see **To cancel key deletion** under [Scheduling and canceling key deletion \(console\)](#) in the *AWS Key Management Service Developer Guide*.

[Lambda.1] Lambda function policies should prohibit public access

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS::Lambda::Function

AWS Config rule: [lambda-function-public-access-prohibited](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the Lambda function resource-based policy prohibits public access outside of your account.

The control also fails if a Lambda function is invoked from Amazon S3 and the policy does not include a condition for `AWS:SourceAccount`.

The Lambda function should not be publicly accessible, as this may allow unintended access to your code stored in the function.

Note

This control is not supported in the China (Beijing) or China (Ningxia) Regions.

Remediation

If a Lambda function fails this control, it indicates that the resource-based policy statement for the Lambda function allows public access.

To remediate the issue, you must update the policy to remove the permissions or to add the `AWS:SourceAccount` condition. You can only update the resource-based policy from the Lambda API.

The following instructions use the console to review the policy and the AWS Command Line Interface to remove the permissions.

To view the resource-based policy for a Lambda function

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. In the navigation pane, choose **Functions**.
3. Choose the function.
4. Choose **Permissions**. The resource-based policy shows the permissions that are applied when another account or AWS service attempts to access the function.
5. Examine the resource-based policy. Identify the policy statement that has `Principal` field values that make the policy public. For example, allowing "*" or { "AWS": "*" }.

You cannot edit the policy from the console. To remove permissions from the function, you use the `remove-permission` command from the AWS CLI.

Note the value of the statement ID (`sid`) for the statement that you want to remove.

To use the AWS CLI to remove permissions from a Lambda function, issue the `remove-permission` command.

```
$ aws lambda remove-permission --function-name <function-name> --statement-id <statement-id>
```

Replace `<function-name>` with the name of the Lambda function, and `<statement-id>` with the statement ID of the statement to remove.

To verify that the permissions are updated

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.
2. In the navigation pane, choose **Functions**.
3. Choose the function that you updated.
4. Choose **Permissions**.

The resource-based policy should be updated. If there was only one statement in the policy, then the policy is empty.

For more information, see [Using resource-based policies for AWS Lambda](#) in the *AWS Lambda Developer Guide*.

[Lambda.2] Lambda functions should use supported runtimes

Category: Protect > Secure development

Severity: Medium

Resource type: AWS::Lambda::Function

AWS Config rule: [lambda-function-settings-check](#)

Schedule type: Change triggered

Parameters:

- `runtime`: `nodejs16.x`, `nodejs14.x`, `nodejs12.x`, `python3.9`, `python3.8`, `python3.7`, `python3.6`, `ruby2.7`, `java11`, `java8`, `java8.al2`, `go1.x`, `dotnetcore3.1`, `dotnet6`

This control checks that the Lambda function settings for runtimes match the expected values set for the supported runtimes for each language. This control checks function settings for the following runtimes: `nodejs16.x`, `nodejs14.x`, `nodejs12.x`, `python3.9`, `python3.8`, `python3.7`, `python3.6`, `ruby2.7`, `java11`, `java8`, `java8.al2`, `go1.x`, `dotnetcore3.1`, and `dotnet6`.

The AWS Config rule ignores functions that have a package type of `Image`.

[Lambda runtimes](#) are built around a combination of operating system, programming language, and software libraries that are subject to maintenance and security updates. When a runtime component is no longer supported for security updates, Lambda deprecates the runtime. Even though you cannot create functions that use the deprecated runtime, the function is still available to process invocation events. Make sure that your Lambda functions are current and do not use out-of-date runtime environments.

To learn more about the supported runtimes that this control checks for the supported languages, see [AWS Lambda runtimes](#) in the *AWS Lambda Developer Guide*.

Note

This control is not supported in the China (Beijing) or China (Ningxia) Regions.

Remediation

For more information on supported runtimes and deprecation schedules, see the [Runtime support policy](#) section of the *AWS Lambda Developer Guide*. When you migrate your runtimes to the latest version, follow the syntax and guidance from the publishers of the language.

[Lambda.4] Lambda functions should have a dead-letter queue configured (Retired)

This control is retired.

[Lambda.5] VPC Lambda functions should operate in more than one Availability Zone

Category: Recover > Resilience > High Availability

Severity: Medium

Resource type: `AWS::Lambda::Function`

AWS Config rule: [lambda-vpc-multi-az-check](#)

Schedule type: Change triggered

Parameters: None

This control checks if Lambda has more than one availability zone associated. The rule fails if only one availability zone is associated with Lambda.

Deploying resources across multiple Availability Zones is an AWS best practice to ensure high availability within your architecture. Availability is a core pillar in the confidentiality, integrity, and availability triad security model. All Lambda functions should have a multi-Availability Zone deployment to ensure that a single zone of failure does not cause a total disruption of operations.

Remediation

To deploy a Lambda function in multiple Availability Zones through console:

1. Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>
2. From the **Functions** page on the Lambda console choose a function.

3. Choose **Configuration** and then choose **VPC**.
4. Choose **Edit**.
5. If the function was not originally connected to a VPC, select a VPC from the dropdown menu. If the function was not originally connected to a VPC, choose at least one security group to attach to the function

Note

The function execution role must have permissions to call `CreateNetworkInterface` on EC2.

6. Choose **Save**.

[Network Firewall.3] Network Firewall policies should have at least one rule group associated

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: `netfw-policy-rule-group-associated`

Schedule type: Change triggered

Parameters: None

This control checks whether a Network Firewall policy has any stateful or stateless rule groups associated. The control fails if stateless or stateful rule groups are not assigned.

A firewall policy defines how your firewall monitors and handles traffic in Amazon Virtual Private Cloud (Amazon VPC). Configuration of stateless and stateful rule groups helps to filter packets and traffic flows, and defines default traffic handling.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To add a rule group to a Network Firewall policy, see [Updating a firewall policy](#) in the *AWS Network Firewall Developer Guide*. For information about creating and managing rule groups, see [Rule groups in AWS Network Firewall](#).

[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: [netfw-policy-default-action-full-packets](#)

Schedule type: Change triggered

Parameters:

- `statelessDefaultActions: aws:drop,aws:forward_to_sfe`

This control checks whether the default stateless action for full packets for a Network Firewall policy is drop or forward. The control passes if `Drop` or `Forward` is selected, and fails if `Pass` is selected.

A firewall policy defines how your firewall monitors and handles traffic in Amazon VPC. You configure stateless and stateful rule groups to filter packets and traffic flows. Defaulting to `Pass` can allow unintended traffic.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To change the firewall policy:

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Network Firewall**, choose **Firewall policies**.
3. Select the name of the firewall policy that you want to edit. This takes you to the firewall policy's details page.
4. In **Stateless Default Actions**, choose **Edit**. Then choose **Drop** or **Forward to stateful rule groups** as the **Default actions for full packets**.

[\[NetworkFirewall.5\] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets](#)

Category: Protect > Secure Network Configuration

Severity: Medium

Resource type: `AWS::NetworkFirewall::FirewallPolicy`

AWS Config rule: [netfw-policy-default-action-fragment-packets](#)

Schedule type: Change triggered

Parameters:

- `statelessFragDefaultActions (Required) : aws:drop, aws:forward_to_sfe`

This control checks whether the default stateless action for fragmented packets for a Network Firewall policy is drop or forward. The control passes if `Drop` or `Forward` is selected, and fails if `Pass` is selected.

A firewall policy defines how your firewall monitors and handles traffic in Amazon VPC. You configure stateless and stateful rule groups to filter packets and traffic flows. Defaulting to `Pass` can allow unintended traffic.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To change the firewall policy:

1. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
2. In the navigation pane, under **Network Firewall**, choose **Firewall policies**.
3. Select the name of the firewall policy that you want to edit. This takes you to the firewall policy's details page.
4. In **Stateless Default Actions**, choose **Edit**. Then choose **Drop** or **Forward to stateful rule groups as the Default actions for fragmented packets**.

[OpenSearch.1] OpenSearch domains should have encryption at rest enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-encrypted-at-rest](#)

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains have encryption-at-rest configuration enabled. The check fails if encryption at rest is not enabled.

For an added layer of security for sensitive data, you should configure your OpenSearch Service domain to be encrypted at rest. When you configure encryption of data at rest, AWS KMS stores and manages your encryption keys. To perform the encryption, AWS KMS uses the Advanced Encryption Standard algorithm with 256-bit keys (AES-256).

To learn more about OpenSearch Service encryption at rest, see [Encryption of data at rest for Amazon OpenSearch Service](#) in the *Amazon OpenSearch Service Developer Guide*.

Remediation

By default, domains do not encrypt data at rest, and you cannot configure existing domains to use the feature. To enable the feature, you must create another domain and migrate your data.

For information about creating domains, see [Creating and managing Amazon OpenSearch Service domains](#) in the *Amazon OpenSearch Service Developer Guide*.

Encryption of data at rest requires Amazon OpenSearch 1.0 or later. For more information about encrypting data at rest for Amazon OpenSearch, see [Encryption of data at rest for Amazon OpenSearch Service](#) in the *Amazon OpenSearch Service Developer Guide*.

[OpenSearch.2] OpenSearch domains should be in a VPC

Category: Protect > Secure network configuration > Resources within VPC

Severity: Critical

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-in-vpc-only](#)

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains are in a VPC. It does not evaluate the VPC subnet routing configuration to determine public access.

You should ensure that OpenSearch domains are not attached to public subnets. See [Resource-based policies](#) in the Amazon OpenSearch Service Developer Guide. You should also ensure that your VPC is configured according to the recommended best practices. See [Security best practices for your VPC](#) in the Amazon VPC User Guide.

OpenSearch domains deployed within a VPC can communicate with VPC resources over the private AWS network, without the need to traverse the public internet. This configuration increases the security posture by limiting access to the data in transit. VPCs provide a number of network controls to secure access to OpenSearch domains, including network ACL and security groups. Security Hub recommends that you migrate public OpenSearch domains to VPCs to take advantage of these controls.

Remediation

If you create a domain with a public endpoint, you cannot later place it within a VPC. Instead, you must create a new domain and migrate your data. The reverse is also true. If you create a domain within a VPC, it cannot have a public endpoint.

Instead, you must either [create another domain](#) or disable this control.

See [Launching your Amazon OpenSearch Service domains within a VPC](#) in the Amazon OpenSearch Service Developer Guide.

[OpenSearch.3] OpenSearch domains should encrypt data sent between nodes

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-node-to-node-encryption-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains have node-to-node encryption enabled. This control fails if node-to-node encryption is disabled on the domain.

HTTPS (TLS) can be used to help prevent potential attackers from eavesdropping on or manipulating network traffic using person-in-the-middle or similar attacks. Only encrypted connections over HTTPS

(TLS) should be allowed. Enabling node-to-node encryption for OpenSearch domains ensures that intra-cluster communications are encrypted in transit.

There can be a performance penalty associated with this configuration. You should be aware of and test the performance trade-off before enabling this option.

Remediation

Node-to-node encryption can only be enabled on a new domain. To remediate this finding, create a new domain with **Node-to-node encryption** enabled and migrate your data to the new domain. Follow the instructions to [create a new domain](#) in the Amazon OpenSearch Service Developer Guide and ensure that you select the **Node-to-node encryption** option when creating the new domain. Then follow [Using a snapshot to migrate data](#) to migrate your data to the new domain.

[OpenSearch.4] OpenSearch domain error logging to CloudWatch Logs should be enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-logs-to-cloudwatch](#)

Schedule type: Change triggered

Parameters:

- `logtype = 'error'`

This control checks whether OpenSearch domains are configured to send error logs to CloudWatch Logs. This control fails if error logging to CloudWatch is not enabled for a domain.

You should enable error logs for OpenSearch domains and send those logs to CloudWatch Logs for retention and response. Domain error logs can assist with security and access audits, and can help to diagnose availability issues.

Remediation

For information on how to enable log publishing, see [Enabling log publishing \(console\)](#) in the Amazon OpenSearch Service Developer Guide.

[OpenSearch.5] OpenSearch domains should have audit logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-audit-logging-enabled](#)

Schedule type: Change triggered

Parameters:

- `cloudWatchLogsLogGroupArnList` (Optional). Security Hub does not populate this parameter. Comma-separated list of CloudWatch Logs log groups that should be configured for audit logs.

This rule is `NON_COMPLIANT` if the CloudWatch Logs log group of the OpenSearch domain is not specified in this parameter list.

This control checks whether OpenSearch domains have audit logging enabled. This control fails if an OpenSearch domain does not have audit logging enabled.

Audit logs are highly customizable. They allow you to track user activity on your OpenSearch clusters, including authentication successes and failures, requests to OpenSearch, index changes, and incoming search queries.

Remediation

For detailed instructions on enabling audit logs, see [Enabling audit logs](#) in the Amazon OpenSearch Service Developer Guide.

[OpenSearch.6] OpenSearch domains should have at least three data nodes

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-data-node-fault-tolerance`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains are configured with at least three data nodes and `zoneAwarenessEnabled` is `true`. This control fails for an OpenSearch domain if `instanceCount` is less than 3 or `zoneAwarenessEnabled` is `false`.

An OpenSearch domain requires at least three data nodes for high availability and fault-tolerance. Deploying an OpenSearch domain with at least three data nodes ensures cluster operations if a node fails.

Remediation

To modify the number of data nodes in an OpenSearch domain

1. Sign in to the AWS console and open the Amazon OpenSearch Service console at <https://console.aws.amazon.com/es/>.
2. Under **My domains**, choose the name of the domain to edit, and choose **Edit**.
3. Under **Data nodes** set **Number of nodes** to a number greater than 3. If you are deploying to three Availability Zone, set the number to a multiple of three to ensure equal distribution across Availability Zones.
4. Choose **Submit**.

[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled

Category: Protect > Secure Access Management > Sensitive API actions restricted

Severity: High

Resource type: `AWS::OpenSearch::Domain`

AWS Config rule: `opensearch-access-control-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether OpenSearch domains have fine-grained access control enabled. The control fails if the fine-grained access control is not enabled. Fine-grained access control requires `advanced-security-options` in the OpenSearch parameter `update-domain-config` to be enabled.

Fine-grained access control offers additional ways of controlling access to your data on Amazon OpenSearch Service.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To enable fine-grained access control, see [Fine-grained access control in Amazon OpenSearch Service](#) in the *Amazon OpenSearch Service Developer Guide*.

[OpenSearch.8] Connections to OpenSearch domains should be encrypted using TLS 1.2

Category: Protect > Data protection > Encryption of data-in-transit

Severity: Medium

Resource type: AWS::OpenSearch::Domain

AWS Config rule: [opensearch-https-required](#)

Schedule type: Change triggered

Parameters: None

This control checks whether connections to OpenSearch domains are required to use TLS 1.2. The check fails if the OpenSearch domain `TLSecurityPolicy` is not `Policy-Min-TLS-1-2-2019-07`.

HTTPS (TLS) can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over HTTPS (TLS) should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS. TLS 1.2 provides several security enhancements over previous versions of TLS.

Remediation

To enable TLS encryption, use the [UpdateDomainConfig](#) API operation to configure the [DomainEndpointOptions](#) in order to set the `TLSecurityPolicy`. For more information, see [Node-to-node encryption](#) in the Amazon OpenSearch Service Developer Guide.

[RDS.1] RDS snapshots should be private

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS::RDS::DBSnapshot, AWS::RDS::DBClusterSnapshot

AWS Config rule: [rds-snapshots-public-prohibited](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon RDS snapshots are public. The control fails if RDS snapshots are public. This control evaluates RDS instances, Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters.

RDS snapshots are used to back up the data on your RDS instances at a specific point in time. They can be used to restore previous states of RDS instances.

An RDS snapshot must not be public unless intended. If you share an unencrypted manual snapshot as public, this makes the snapshot available to all AWS accounts. This may result in unintended data exposure of your RDS instance.

Note that if the configuration is changed to allow public access, the AWS Config rule may not be able to detect the change for up to 12 hours. Until the AWS Config rule detects the change, the check passes even though the configuration violates the rule.

To learn more about sharing a DB snapshot, see [Sharing a DB snapshot](#) in the *Amazon RDS User Guide*.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

To remediate this issue, update your RDS snapshots to remove public access.

To remove public access for RDS snapshots

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Navigate to **Snapshots** and then choose the public snapshot you want to modify.
3. From **Actions**, choose **Share Snapshots**.
4. From **DB snapshot visibility**, choose **Private**.
5. Under **DB snapshot visibility**, choose **all**.
6. Choose **Save**.

[RDS.2] Amazon RDS DB instances should prohibit public access, determined by the PubliclyAccessible configuration

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-instance-public-access-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon RDS instances are publicly accessible by evaluating the `PubliclyAccessible` field in the instance configuration item.

Neptune DB instances and Amazon DocumentDB clusters do not have the `PubliclyAccessible` flag and cannot be evaluated. However, this control can still generate findings for these resources. You can suppress these findings.

The `PubliclyAccessible` value in the RDS instance configuration indicates whether the DB instance is publicly accessible. When the DB instance is configured with `PubliclyAccessible`, it is an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. When the DB instance isn't publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address.

Unless you intend for your RDS instance to be publicly accessible, the RDS instance should not be configured with `PubliclyAccessible` value. Doing so might allow unnecessary traffic to your database instance.

Remediation

To remediate this issue, update your RDS DB instances to remove public access.

To remove public access from RDS DB instances

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Navigate to **Databases** and then choose your public database.
3. Choose **Modify**.
4. Under **Connectivity**, expand **Additional connectivity configuration**.
5. Under **Public access**, choose **Not publicly accessible**.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose **Apply immediately**.
8. Choose **Modify DB Instance**.

For more information, see [Working with a DB instance in a VPC](#) in the *Amazon RDS User Guide*.

[RDS.3] RDS DB instances should have encryption at rest enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-storage-encrypted](#)

Schedule type: Change triggered

Parameters: None

This control checks whether storage encryption is enabled for your Amazon RDS DB instances.

This control is intended for RDS DB instances. However, it can also generate findings for Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters. If these findings are not useful, then you can suppress them.

For an added layer of security for your sensitive data in RDS DB instances, you should configure your RDS DB instances to be encrypted at rest. To encrypt your RDS DB instances and snapshots at rest, enable

the encryption option for your RDS DB instances. Data that is encrypted at rest includes the underlying storage for DB instances, its automated backups, read replicas, and snapshots.

RDS encrypted DB instances use the open standard AES-256 encryption algorithm to encrypt your data on the server that hosts your RDS DB instances. After your data is encrypted, Amazon RDS handles authentication of access and decryption of your data transparently with a minimal impact on performance. You do not need to modify your database client applications to use encryption.

Amazon RDS encryption is currently available for all database engines and storage types. Amazon RDS encryption is available for most DB instance classes. To learn about DB instance classes that do not support Amazon RDS encryption, see [Encrypting Amazon RDS resources](#) in the *Amazon RDS User Guide*.

Remediation

For information about encrypting DB instances in Amazon RDS, see [Encrypting Amazon RDS resources](#) in the *Amazon RDS User Guide*.

[RDS.4] RDS cluster snapshots and database snapshots should be encrypted at rest

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::RDS::DBClusterSnapshot, AWS::RDS::DBSnapshot

AWS Config rule: [rds-snapshots-encrypted](#)

Schedule type: Change triggered

Parameters: None

This control checks whether RDS DB snapshots are encrypted.

This control is intended for RDS DB instances. However, it can also generate findings for snapshots of Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters. If these findings are not useful, then you can suppress them.

Encrypting data at rest reduces the risk that an unauthenticated user gets access to data that is stored on disk. Data in RDS snapshots should be encrypted at rest for an added layer of security.

Remediation

You can use the Amazon RDS console to remediate this issue.

To encrypt an unencrypted RDS snapshot

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Snapshots**.
3. Find the snapshot to encrypt under **Manual** or **System**.
4. Select the check box next to the snapshot to encrypt.
5. Choose **Actions**, then choose **Copy Snapshot**.
6. Under **New DB Snapshot Identifier**, type a name for the new snapshot.
7. Under **Encryption**, select **Enable Encryption**.
8. Choose the KMS key to use to encrypt the snapshot.
9. Choose **Copy Snapshot**.
10. After the new snapshot is created, delete the original snapshot.
11. For **Backup Retention Period**, choose a positive nonzero value. For example, 30 days.

[RDS.5] RDS DB instances should be configured with multiple Availability Zones

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-multi-az-support](#)

Schedule type: Change triggered

Parameters: None

This control checks whether high availability is enabled for your RDS DB instances.

RDS DB instances should be configured for multiple Availability Zones (AZs). This ensures the availability of the data stored. Multi-AZ deployments allow for automated failover if there is an issue with Availability Zone availability and during regular RDS maintenance.

Remediation

To remediate this issue, update your DB instances to enable multiple Availability Zones.

To enable multiple Availability Zones for a DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify**. The **Modify DB Instance** page appears.
4. Under **Instance Specifications**, set **Multi-AZ deployment** to **Yes**.
5. Choose **Continue** and then check the summary of modifications.
6. (Optional) Choose **Apply immediately** to apply the changes immediately. Choosing this option can cause an outage in some cases. For more information, see [Using the Apply Immediately setting](#) in the *Amazon RDS User Guide*.
7. On the confirmation page, review your changes. If they are correct, choose **Modify DB Instance** to save your changes.

[RDS.6] Enhanced monitoring should be configured for RDS DB instances and clusters

Category: Detect > Detection services

Severity: Low

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-enhanced-monitoring-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether enhanced monitoring is enabled for your RDS DB instances.

In Amazon RDS, Enhanced Monitoring enables a more rapid response to performance changes in underlying infrastructure. These performance changes could result in a lack of availability of the data. Enhanced Monitoring provides real-time metrics of the operating system that your RDS DB instance runs

on. An agent is installed on the instance. The agent can obtain metrics more accurately than is possible from the hypervisor layer.

Enhanced Monitoring metrics are useful when you want to see how different processes or threads on a DB instance use the CPU. For more information, see [Enhanced Monitoring](#) in the *Amazon RDS User Guide*.

Remediation

For detailed instructions on how to enable Enhanced Monitoring for your DB instance, see [Setting up for and enabling Enhanced Monitoring](#) in the *Amazon RDS User Guide*.

[RDS.7] RDS clusters should have deletion protection enabled

Category: Protect > Data protection > Data deletion protection

Severity: Low

Resource type: AWS::RDS::DBCluster

AWS Config rule: `rds-cluster-deletion-protection-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether RDS clusters have deletion protection enabled.

This control is intended for RDS DB instances. However, it can also generate findings for Aurora DB instances, Neptune DB instances, and Amazon DocumentDB clusters. If these findings are not useful, then you can suppress them.

Enabling cluster deletion protection is an additional layer of protection against accidental database deletion or deletion by an unauthorized entity.

When deletion protection is enabled, an RDS cluster cannot be deleted. Before a deletion request can succeed, deletion protection must be disabled.

Note

This control is not supported in the following Regions:

- China (Beijing)
- China (Ningxia)
- Middle East (Bahrain)
- South America (São Paulo).

Remediation

To remediate this issue, update your RDS DB cluster to enable delete protection.

To enable deletion protection for an RDS DB cluster

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, then choose the DB cluster that you want to modify.
3. Choose **Modify**.
4. Under **Deletion protection**, choose **Enable deletion protection**.
5. Choose **Continue**.
6. Under **Scheduling of modifications**, choose when to apply modifications. The options are **Apply during the next scheduled maintenance window** or **Apply immediately**.

7. Choose **Modify Cluster**.

[RDS.8] RDS DB instances should have deletion protection enabled

Category: Protect > Data protection > Data deletion protection

Severity: Low

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-instance-deletion-protection-enabled](#)

Schedule type: Change triggered

Parameters:

- `databaseEngines: mariadb,mysql,oracle-ee,oracle-se2,oracle-se1,oracle-se,postgres,sqlserver-ee,sqlserver-se,sqlserver-ex,sqlserver-web`

This control checks whether your RDS DB instances that use one of the listed database engines have deletion protection enabled.

Enabling instance deletion protection is an additional layer of protection against accidental database deletion or deletion by an unauthorized entity.

While deletion protection is enabled, an RDS DB instance cannot be deleted. Before a deletion request can succeed, deletion protection must be disabled.

Remediation

To remediate this issue, update your RDS DB instance to enable deletion protection.

To enable deletion protection for an RDS DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, then choose the DB instance that you want to modify.
3. Choose **Modify**.
4. Under **Deletion protection**, choose **Enable deletion protection**.
5. Choose **Continue**.
6. Under **Scheduling of modifications**, choose when to apply modifications. The options are **Apply during the next scheduled maintenance window** or **Apply immediately**.
7. Choose **Modify DB Instance**.

[RDS.9] Database logging should be enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-logging-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the following logs of Amazon RDS are enabled and sent to CloudWatch Logs:

- Oracle: (Alert, Audit, Trace, Listener)
- PostgreSQL: (Postgresql, Upgrade)
- MySQL: (Audit, Error, General, SlowQuery)
- MariaDB: (Audit, Error, General, SlowQuery)
- SQL Server: (Error, Agent)
- Aurora: (Audit, Error, General, SlowQuery)
- Aurora-MySQL: (Audit, Error, General, SlowQuery)
- Aurora-PostgreSQL: (Postgresql, Upgrade).

RDS databases should have relevant logs enabled. Database logging provides detailed records of requests made to RDS. Database logs can assist with security and access audits and can help to diagnose availability issues.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Osaka)
- China (Ningxia)
- Europe (Milan)

Remediation

Logging options are contained in the DB parameter group associated with the RDS DB cluster or instance. To enable logging when the default parameter group for the database engine is used, you must create a new DB parameter group that has the required parameter values. You must then associate the customer DB parameter group with the DB cluster or instance.

To enable and publish MariaDB, MySQL, or PostgreSQL logs to CloudWatch Logs from the AWS Management Console, set the following parameters in a custom DB Parameter Group:

Database engine	Parameters
MariaDB	<pre>general_log=1 slow_query_log=1 log_output = FILE</pre> <p>MariaDB also requires a custom options group, explained below.</p>
MySQL	<pre>general_log=1 slow_query_log=1 log_output = FILE</pre>
PostgreSQL	<pre>log_statement=all log_min_duration_statement=<i>minimum query duration (ms) to log</i></pre>

To create a custom DB parameter group

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Parameter groups**.
3. Choose **Create parameter group**. The **Create parameter group** window appears.
4. In the **Parameter group** family list, choose a DB parameter group family.
5. In the **Type** list, choose **DB Parameter Group**.
6. In **Group name**, enter the name of the new DB parameter group.
7. In **Description**, enter a description for the new DB parameter group.
8. Choose **Create**.

To create a new option group for MariaDB logging by using the console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Option groups**.
3. Choose **Create group**.
4. In the **Create option group** window, do the following:
 - a. For **Name**, type a name for the option group that is unique within your AWS account. The name can contain only letters, digits, and hyphens.
 - b. For **Description**, type a brief description of the option group. The description is used for display purposes.
 - c. For **Engine**, choose the DB engine that you want.
 - d. For **Major engine version**, choose the major version of the DB engine that you want.
5. To continue, choose **Create**.
6. Choose the name of the option group you just created.
7. Choose **Add option**.
8. Choose **MARIADB_AUDIT_PLUGIN** from the **Option name** list.
9. Set `SERVER_AUDIT_EVENTS` to `CONNECT`, `QUERY`, `TABLE`, `QUERY_DDL`, `QUERY_DML`, `QUERY_DCL`.
10. Choose **Add option**.

To publish SQL Server DB, Oracle DB, or PostgreSQL logs to CloudWatch Logs from the AWS Management Console

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.
3. Choose the DB instance that you want to modify.
4. Choose **Modify**.
5. Under **Log exports**, choose all of the log files to start publishing to CloudWatch Logs.

Log exports is available only for database engine versions that support publishing to CloudWatch Logs.
6. Choose **Continue**. Then on the summary page, choose **Modify DB Instance**.

To apply a new DB parameter group or DB options group to an RDS DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**.

3. Choose the DB instance that you want to modify.
4. Choose **Modify**. The **Modify DB Instance** page appears.
5. Under **Database options**, change the DB parameter group and DB options group as needed.
6. When you finish your changes, choose **Continue**. Check the summary of modifications.
7. (Optional) Choose **Apply immediately** to apply the changes immediately. Choosing this option can cause an outage in some cases. For more information, see [Using the Apply Immediately setting](#) in the *Amazon RDS User Guide*.
8. Choose **Modify DB Instance** to save your changes.

[RDS.10] IAM authentication should be configured for RDS instances

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-instance-iam-authentication-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an RDS DB instance has IAM database authentication enabled.

IAM database authentication allows authentication to database instances with an authentication token instead of a password. Network traffic to and from the database is encrypted using SSL. For more information, see [IAM database authentication](#) in the *Amazon Aurora User Guide*.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)

Remediation

To remediate this issue, update your DB instance to enable IAM authentication.

To enable IAM authentication for an existing DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Select the DB instance to modify.
4. Choose **Modify**.
5. Under **Database options**, choose **Enable IAM DB authentication**.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose when to apply modifications. The options are **Apply during the next scheduled maintenance window** or **Apply immediately**.
8. For clusters, choose **Modify DB Instance**.

[RDS.11] Amazon RDS instances should have automatic backups enabled

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: AWS::RDS::DBCluster

AWS Config rule: [db-instance-backup-enabled](#)

Schedule type: Change triggered

Parameters:

- `backupRetentionMinimum: 7`

This control checks whether Amazon Relational Database Service instances have automated backups enabled and the backup retention period is greater than or equal to seven days. The control fails if backups are not enabled, and if the retention period is less than 7 days.

Backups help you more quickly recover from a security incident and strengthens the resilience of your systems. Amazon RDS provides an easy way to configure daily full instance volume snapshots. For more details on Amazon RDS automated backups, see [Working with Backups](#) in the Amazon RDS User Guide.

Note

This control is not supported in Asia Pacific (Osaka).

Remediation

To enable automated backups immediately

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance that you want to modify.
3. Choose **Modify** to open the **Modify DB Instance** page.
4. Under **Backup Retention Period**, choose a positive nonzero value, for example 30 days, then choose **Continue**.
5. Select the **Scheduling of modifications** section and choose when to apply modifications: you can choose to **Apply during the next scheduled maintenance window** or **Apply immediately**.
6. Then, on the confirmation page, choose **Modify DB Instance** to save your changes and enable automated backups.

[RDS.12] IAM authentication should be configured for RDS clusters

Category: Protect > Secure access management > Passwordless authentication

Severity: Medium

Resource type: AWS::RDS::DBCluster, AWS::RDS::DBInstance

AWS Config rule: [rds-cluster-iam-authentication-enabled](#)

Schedule type: Change triggered

Parameters:None

This control checks whether an Amazon RDS DB cluster has IAM database authentication enabled.

IAM database authentication allows for password-free authentication to database instances. The authentication uses an authentication token. Network traffic to and from the database is encrypted using SSL. For more information, see [IAM database authentication](#) in the *Amazon Aurora User Guide*.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Middle East (Bahrain)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can enable IAM authentication for a DB cluster from the Amazon RDS console.

To enable IAM authentication for an existing DB cluster

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Choose the DB cluster to modify.
4. Choose **Modify**.
5. Under **Database options**, select **Enable IAM DB authentication**.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose when to apply modifications: **Apply during the next scheduled maintenance window** or **Apply immediately**.
8. Choose **Modify cluster**.

[RDS.13] RDS automatic minor version upgrades should be enabled

Category: Detect > Vulnerability and patch management

Severity: High

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-automatic-minor-version-upgrade-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether automatic minor version upgrades are enabled for the RDS database instance.

Enabling automatic minor version upgrades ensures that the latest minor version updates to the relational database management system (RDBMS) are installed. These upgrades might include security patches and bug fixes. Keeping up to date with patch installation is an important step in securing systems.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can enable minor version upgrades for a DB instance from the Amazon RDS console.

To enable automatic minor version upgrades for an existing DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Choose the DB instance to modify.
4. Choose **Modify**.
5. Under **Maintenance**, select **Yes** for **Auto minor version upgrade**.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose when to apply modifications: **Apply during the next scheduled maintenance window** or **Apply immediately**.
8. Choose **Modify DB Instance**.

[RDS.14] Amazon Aurora clusters should have backtracking enabled

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `aurora-mysql-backtracking-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon Aurora clusters have backtracking enabled.

Backups help you to recover more quickly from a security incident. They also strengthen the resilience of your systems. Aurora backtracking reduces the time to recover a database to a point in time. It does not require a database restore to do so.

For more information about backtracking in Aurora, see [Backtracking an Aurora DB cluster](#) in the *Amazon Aurora User Guide*.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)

- Europe (Stockholm)
- Middle East (Bahrain)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For detailed instructions on how to enable Aurora backtracking, see [Configuring backtracking](#) in the *Amazon Aurora User Guide*.

Note that you cannot enable backtracking on an existing cluster. Instead, you can create a clone that has backtracking enabled. For more information about the limitations of Aurora backtracking, see the list of limitations in [Overview of backtracking](#).

For information about pricing for backtracking, see the [Aurora pricing page](#).

[RDS.15] RDS DB clusters should be configured for multiple Availability Zones

Category: Recover > Resilience > High availability

Severity: Medium

Resource type: `AWS::RDS::DBCluster`

AWS Config rule: `rds-cluster-multi-az-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether high availability is enabled for your RDS DB clusters.

RDS DB clusters should be configured for multiple Availability Zones to ensure availability of the data that is stored. Deployment to multiple Availability Zones allows for automated failover in the event of an Availability Zone availability issue and during regular RDS maintenance events.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- Middle East (Bahrain)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To remediate this control, configure your DB cluster for multiple Availability Zones.

To enable multi-AZ for a DB cluster

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Databases**, and then choose the DB instance to modify.

3. Choose **Modify**. The **Modify DB Instance** page appears.
4. Under **Instance Specifications**, set **Multi-AZ deployment** to **Yes**.
5. Choose **Continue** and check the summary of modifications.
6. (Optional) Choose **Apply immediately** to apply the changes immediately. Choosing this option can cause an outage in some cases. For more information, see [Using the Apply Immediately setting](#) in the *Amazon RDS User Guide*.

On the confirmation page, review your changes. If they are correct, choose **Modify DB Instance**.

[RDS.16] RDS DB clusters should be configured to copy tags to snapshots

Category: Identify > Inventory

Severity: Low

Resource type: AWS::RDS::DBCluster

AWS Config rule: `rds-cluster-copy-tags-to-snapshots-enabled` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether RDS DB clusters are configured to copy all tags to snapshots when the snapshots are created.

Identification and inventory of your IT assets is a crucial aspect of governance and security. You need to have visibility of all your RDS DB clusters so that you can assess their security posture and take action on potential areas of weakness. Snapshots should be tagged in the same way as their parent RDS database clusters. Enabling this setting ensures that snapshots inherit the tags of their parent database clusters.

Note

This control is not supported in the following Regions:

- China (Beijing)
- Middle East (Bahrain)
- South America (São Paulo)

Remediation

To enable automatic tag copying to snapshots for a DB cluster

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Select the DB cluster to modify.
4. Choose **Modify**.
5. Under **Backup**, select **Copy tags to snapshots**.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose when to apply modifications. You can choose either **Apply during the next scheduled maintenance window** or **Apply immediately**.

[RDS.17] RDS DB instances should be configured to copy tags to snapshots

Category: Identify > Inventory

Severity: Low

Resource type: AWS::RDS::DBInstance

AWS Config rule: `rds-instance-copy-tags-to-snapshots-enabled` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether RDS DB instances are configured to copy all tags to snapshots when the snapshots are created.

Identification and inventory of your IT assets is a crucial aspect of governance and security. You need to have visibility of all your RDS DB instances so that you can assess their security posture and take action on potential areas of weakness. Snapshots should be tagged in the same way as their parent RDS database instances. Enabling this setting ensures that snapshots inherit the tags of their parent database instances.

Remediation

To enable automatic tag copying to snapshots for a DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Select the DB instance to modify.
4. Choose **Modify**.
5. Under **Backup**, select **Copy tags to snapshots**.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose when to apply modifications. You can choose either **Apply during the next scheduled maintenance window** or **Apply immediately**.

[RDS.18] RDS instances should be deployed in a VPC

Category: Protect > Secure network configuration > Resources within VPC

Severity: High

Resource type: AWS::RDS::DBInstance

AWS Config rule: `rds-deployed-in-vpc` (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether an RDS instance is deployed in a VPC (EC2-VPC).

VPCs provide a number of network controls to secure access to RDS resources. These controls include VPC Endpoints, network ACLs, and security groups. To take advantage of these controls, we recommend that you move EC2-Classic RDS instances to EC2-VPC.

Remediation

For detailed instructions on how to move RDS instances to VPC, see [Updating the VPC for a DB instance](#) in the *Amazon RDS User Guide*.

[RDS.19] An RDS event notifications subscription should be configured for critical cluster events

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: AWS::RDS::EventSubscription

AWS Config rule: rds-cluster-event-notifications-configured (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS event subscription exists that has notifications enabled for the following source type, event category key-value pairs.

```
DBCluster: ["maintenance", "failure"]
```

RDS event notifications uses Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response. For additional information about RDS event notifications, see [Using Amazon RDS event notification](#) in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS cluster event notifications

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Event subscriptions**.
3. Under **Event subscriptions**, choose **Create event subscription**.
4. In the **Create event subscription** dialog, do the following:
 - a. For **Name**, enter a name for the event notification subscription.
 - b. For **Send notifications to**, choose an existing Amazon SNS ARN for an SNS topic. To use a new topic, choose **create topic** to enter the name of a topic and a list of recipients.
 - c. For **Source type**, choose **Clusters**.
 - d. Under **Instances to include**, select **All clusters**.
 - e. Under **Event categories to include**, select **Specific event categories**. The control also passes if you select **All event categories**.
 - f. Select **maintenance** and **failure**.
 - g. Choose **Create**.

[RDS.20] An RDS event notifications subscription should be configured for critical database instance events

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: AWS::RDS::EventSubscription

AWS Config rule: rds-instance-event-notifications-configured (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS event subscription exists with notifications enabled for the following source type, event category key-value pairs.

```
DBInstance: ["maintenance","configuration change","failure"]
```

RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response. For additional information about RDS event notifications, see [Using Amazon RDS event notification](#) in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS instance event notifications

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Event subscriptions**.
3. Under **Event subscriptions**, choose **Create event subscription**.
4. In the **Create event subscription** dialog, do the following:
 - a. For **Name**, enter a name for the event notification subscription.
 - b. For **Send notifications to**, choose an existing Amazon SNS ARN for an SNS topic. To use a new topic, choose **create topic** to enter the name of a topic and a list of recipients.
 - c. For **Source type**, choose **Instances**.
 - d. Under **Instances to include**, select **All instances**.
 - e. Under **Event categories to include**, select **Specific event categories**. The control also passes if you select **All event categories**.
 - f. Select **maintenance**, **configuration change**, and **failure**.
 - g. Choose **Create**.

[RDS.21] An RDS event notifications subscription should be configured for critical database parameter group events

Category: Detect > Detection services > Application monitoring

Severity: Low

Resource type: AWS::RDS::EventSubscription

AWS Config rule: rds-pg-event-notifications-configured (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS event subscription exists with notifications enabled for the following source type, event category key-value pairs.

```
DBParameterGroup: ["configuration change"]
```

RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for rapid response. For additional information about RDS event notifications, see [Using Amazon RDS event notification](#) in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS database parameter group event notifications

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. In the navigation pane, choose **Event subscriptions**.
3. Under **Event subscriptions**, choose **Create event subscription**.
4. In the **Create event subscription** dialog, do the following:
 - a. For **Name**, enter a name for the event notification subscription.
 - b. For **Send notifications to**, choose an existing Amazon SNS ARN for an SNS topic. To use a new topic, choose **create topic** to enter the name of a topic and a list of recipients.
 - c. For **Source type**, choose **Parameter groups**.
 - d. Under **Instances to include**, select **All parameter groups**.
 - e. Under **Event categories to include**, select **Specific event categories**. The control also passes if you select **All event categories**.
 - f. Select **configuration change**.
 - g. Choose **Create**.

[RDS.22] An RDS event notifications subscription should be configured for critical database security group events

Category: Detect > Detection Services > Application monitoring

Severity: Low

Resource type: AWS::RDS::EventSubscription

AWS Config rule: rds-sg-event-notifications-configured (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS event subscription exists with notifications enabled for the following source type, event category key-value pairs.

```
DBSecurityGroup: ["configuration change","failure"]
```

RDS event notifications use Amazon SNS to make you aware of changes in the availability or configuration of your RDS resources. These notifications allow for a rapid response. For additional information about RDS event notifications, see [Using Amazon RDS event notification](#) in the *Amazon RDS User Guide*.

Remediation

To subscribe to RDS database security group event notifications

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.

2. In the navigation pane, choose **Event subscriptions**.
3. Under **Event subscriptions**, choose **Create event subscription**.
4. In the **Create event subscription** dialog, do the following:
 - a. For **Name**, enter a name for the event notification subscription.
 - b. For **Send notifications to**, choose an existing Amazon SNS ARN for an SNS topic. To use a new topic, choose **create topic** to enter the name of a topic and a list of recipients.
 - c. For **Source type**, choose **Security groups**.
 - d. Under **Instances to include**, select **All security groups**.
 - e. Under **Event categories to include**, select **Specific event categories**. The control also passes if you select **All event categories**.
 - f. Select **configuration change** and **failure**.
 - g. Choose **Create**.

[RDS.23] RDS databases and clusters should not use a database engine default port

Category: Protect > Secure network configuration

Severity: Low

Resource type: AWS::RDS::DBInstance

AWS Config rule: rds-no-default-ports (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether the RDS cluster or instance uses a port other than the default port of the database engine.

If you use a known port to deploy an RDS cluster or instance, an attacker can guess information about the cluster or instance. The attacker can use this information in conjunction with other information to connect to an RDS cluster or instance or gain additional information about your application.

When you change the port, you must also update the existing connection strings that were used to connect to the old port. You should also check the security group of the DB instance to ensure that it includes an ingress rule that allows connectivity on the new port.

Remediation

To modify the default port of an existing DB instance

1. Open the Amazon RDS console at <https://console.aws.amazon.com/rds/>.
2. Choose **Databases**.
3. Select the DB instance to modify
4. Choose **Modify**.
5. Under **Database options**, change **Database port** to a non-default value.
6. Choose **Continue**.
7. Under **Scheduling of modifications**, choose when to apply modifications. You can choose either **Apply during the next scheduled maintenance window** or **Apply immediately**.
8. For clusters, choose **Modify cluster**. For instances, choose **Modify DB Instance**.

[RDS.24] RDS database clusters should use a custom administrator username

Category: Identify > Resource Configuration

Severity: Medium

Resource type: AWS::RDS:DBCluster

AWS Config rule: [rds-cluster-default-admin-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon RDS database cluster has changed the admin username from its default value. This rule will fail if the admin username is set to the default value.

When creating an Amazon RDS database, you should change the default admin username to a unique value. Default usernames are public knowledge and should be changed during RDS database creation. Changing the default usernames reduces the risk of unintended access.

Remediation

For changing the admin username associated with the Amazon RDS database cluster, [create a new RDS database cluster](#) and change the default admin username while creating the database.

[RDS.25] RDS database instances should use a custom administrator username

Category: Identify > Resource Configuration

Severity: Medium

Resource type: AWS::RDS::DBInstance

AWS Config rule: [rds-instance-default-admin-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether you've changed the administrative username for Amazon Relational Database Service (Amazon RDS) database instances from the default value. The control fails if the administrative username is set to the default value.

Default administrative usernames on Amazon RDS databases are public knowledge. When creating an Amazon RDS database, you should change the default administrative username to a unique value to reduce the risk of unintended access.

Remediation

To change the administrative username associated with an RDS database instance, first [create a new RDS database instance](#). Change the default administrative username while creating the database.

[Redshift.1] Amazon Redshift clusters should prohibit public access

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Critical

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-cluster-public-access-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon Redshift clusters are publicly accessible. It evaluates the `PubliclyAccessible` field in the cluster configuration item.

The `PubliclyAccessible` attribute of the Amazon Redshift cluster configuration indicates whether the cluster is publicly accessible. When the cluster is configured with `PubliclyAccessible` set to `true`, it is an Internet-facing instance that has a publicly resolvable DNS name, which resolves to a public IP address.

When the cluster is not publicly accessible, it is an internal instance with a DNS name that resolves to a private IP address. Unless you intend for your cluster to be publicly accessible, the cluster should not be configured with `PubliclyAccessible` set to `true`.

Remediation

To remediate this issue, update your Amazon Redshift cluster to disable public access.

To disable public access to an Amazon Redshift cluster

1. Open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation menu, choose **Clusters**, then choose the name of the cluster with the security group to modify.
3. Choose **Actions**, then choose **Modify publicly accessible setting**.
4. Under **Allow instances and devices outside the VPC to connect to your database through the cluster endpoint**, choose **No**.
5. Choose **Confirm**.

[Redshift.2] Connections to Amazon Redshift clusters should be encrypted in transit

Category: Protect > Data protection > Encryption of data in transit

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-require-tls-ssl](#)

Schedule type: Change triggered

Parameters: None

This control checks whether connections to Amazon Redshift clusters are required to use encryption in transit. The check fails if the Amazon Redshift cluster parameter `require_ssl` is not set to 1.

TLS can be used to help prevent potential attackers from using person-in-the-middle or similar attacks to eavesdrop on or manipulate network traffic. Only encrypted connections over TLS should be allowed. Encrypting data in transit can affect performance. You should test your application with this feature to understand the performance profile and the impact of TLS.

Note

This control is not supported in Europe (Milan).

Remediation

To remediate this issue, update the parameter group to require encryption.

To modify a parameter group

1. Open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation menu, choose **Config**, then choose **Workload management** to display the **Workload management** page.
3. Choose the parameter group that you want to modify.
4. Choose **Parameters**.
5. Choose **Edit parameters** then set `require_ssl` to 1.
6. Enter your changes and then choose **Save**.

[Redshift.3] Amazon Redshift clusters should have automatic snapshots enabled

Category: Recover > Resilience > Backups enabled

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-backup-enabled](#)

Schedule type: Change triggered

Parameters:

- `MinRetentionPeriod = 7`

This control checks whether Amazon Redshift clusters have automated snapshots enabled. It also checks whether the snapshot retention period is greater than or equal to seven.

Backups help you to recover more quickly from a security incident. They strengthen the resilience of your systems. Amazon Redshift takes periodic snapshots by default. This control checks whether automatic snapshots are enabled and retained for at least seven days. For more details on Amazon Redshift automated snapshots, see [Automated snapshots](#) in the *Amazon Redshift Cluster Management Guide*.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Asia Pacific (Osaka)
- Asia Pacific (Sydney)
- China (Ningxia)
- Europe (Milan)

Remediation

To remediate this issue, update the snapshot retention period to at least 7.

To modify the snapshot retention period

1. Open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation menu, choose **Clusters**, then choose the name of the cluster to modify.
3. Choose **Edit**.
4. Under **Backup**, set **Snapshot retention** to a value of 7 or greater.

5. Choose **Modify Cluster**.

[Redshift.4] Amazon Redshift clusters should have audit logging enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: redshift-cluster-audit-logging-enabled (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters:

- loggingEnabled = true

This control checks whether an Amazon Redshift cluster has audit logging enabled.

Amazon Redshift audit logging provides additional information about connections and user activities in your cluster. This data can be stored and secured in Amazon S3 and can be helpful in security audits and investigations. For more information, see [Database audit logging](#) in the *Amazon Redshift Cluster Management Guide*.

Remediation

To enable cluster audit logging

1. Open the Amazon Redshift console at <https://console.aws.amazon.com/redshift/>.
2. In the navigation menu, choose **Clusters**, then choose the name of the cluster to modify.
3. Choose **Maintenance and monitoring**.
4. Under **Audit logging**, choose **Edit**.
5. Set **Enable audit logging** to **yes**, then enter the log destination bucket details.
6. Choose **Confirm**.

[Redshift.6] Amazon Redshift should have automatic upgrades to major versions enabled

Category: Detect > Vulnerability and patch management

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: redshift-cluster-maintenancesettings-check

Schedule type: Change triggered

Parameters:

- allowVersionUpgrade = true

This control checks whether automatic major version upgrades are enabled for the Amazon Redshift cluster.

Enabling automatic major version upgrades ensures that the latest major version updates to Amazon Redshift clusters are installed during the maintenance window. These updates might include security patches and bug fixes. Keeping up to date with patch installation is an important step in securing systems.

Note

This control is not supported in Middle East (Bahrain).

Remediation

To remediate this issue from the AWS CLI, use the Amazon Redshift `modify-cluster` command to set the `--allow-version-upgrade` attribute.

```
aws redshift modify-cluster --cluster-identifier clustername --allow-version-upgrade
```

Where *clustername* is the name of your Amazon Redshift cluster.

[Redshift.7] Amazon Redshift clusters should use enhanced VPC routing

Category: Protect > Secure network configuration > API private access

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-enhanced-vpc-routing-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Redshift cluster has `EnhancedVpcRouting` enabled.

Enhanced VPC routing forces all `COPY` and `UNLOAD` traffic between the cluster and data repositories to go through your VPC. You can then use VPC features such as security groups and network access control lists to secure network traffic. You can also use VPC Flow Logs to monitor network traffic.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For detailed remediation instructions, see [Enabling enhanced VPC routing](#) in the *Amazon Redshift Cluster Management Guide*.

[Redshift.8] Amazon Redshift clusters should not use the default Admin username

Category: Identify > Resource Configuration

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-default-admin-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Redshift cluster has changed the admin username from its default value. This control will fail if the admin username for a Redshift cluster is set to `awsuser`.

When creating a Redshift cluster, you should change the default admin username to a unique value. Default usernames are public knowledge and should be changed upon configuration. Changing the default usernames reduces the risk of unintended access.

Remediation

You can't change the admin username for your Amazon Redshift cluster after it is created. To create a new cluster, follow the instructions [here](#).

[Redshift.9] Redshift clusters should not use the default database name

Category: Identify > Resource Configuration

Severity: Medium

Resource type: AWS::Redshift::Cluster

AWS Config rule: [redshift-default-db-name-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an Amazon Redshift cluster has changed the database name from its default value. The control will fail if the database name for a Redshift cluster is set to `dev`.

When creating a Redshift cluster, you should change the default database name to a unique value. Default names are public knowledge and should be changed upon configuration. As an example, a well-known name could lead to inadvertent access if it was used in IAM policy conditions.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can't change the database name for your Amazon Redshift cluster after it is created. For instructions on creating a new cluster, see [Getting started with Amazon Redshift](#) in the *Amazon Redshift Getting Started Guide*.

[S3.1] S3 Block Public Access setting should be enabled

Category: Protect > Secure network configuration

Severity: Medium

Resource type: AWS account

AWS Config rule: [s3-account-level-public-access-blocks-periodic](#)

Schedule type: Periodic

Parameters:

- `ignorePublicAcls: true`
- `blockPublicPolicy: true`
- `blockPublicAcls: true`
- `restrictPublicBuckets: true`

This control checks whether the following Amazon S3 public access block settings are configured at the account level:

- `ignorePublicAcls: true`
- `blockPublicPolicy: true`
- `blockPublicAcls: true`
- `restrictPublicBuckets: true`

The control passes if all of the public access block settings are set to `true`.

The control fails if any of the settings are set to `false`, or if any of the settings are not configured.

Amazon S3 public access block is designed to provide controls across an entire AWS account or at the individual S3 bucket level to ensure that objects never have public access. Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.

Unless you intend to have your S3 buckets be publicly accessible, you should configure the account level Amazon S3 Block Public Access feature.

To learn more, see [Using Amazon S3 Block Public Access](#) in the *Amazon Simple Storage Service User Guide*.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Europe (Milan)
- Middle East (Bahrain)

Remediation

To remediate this issue, enable Amazon S3 Block Public Access.

To enable Amazon S3 Block Public Access

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose **Block public access (account settings)**.
3. Choose **Edit**.
4. Select **Block all public access**.

5. Choose **Save changes**.

For more information, see [Using Amazon S3 block public access](#) in the *Amazon Simple Storage Service User Guide*.

[S3.2] S3 buckets should prohibit public read access

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-bucket-public-read-prohibited](#)

Schedule type: Periodic and change triggered

Parameters: None

This control checks whether your S3 buckets allow public read access. It evaluates the Block Public Access settings, the bucket policy, and the bucket access control list (ACL).

Some use cases require that everyone on the internet be able to read from your S3 bucket. However, those situations are rare. To ensure the integrity and security of your data, your S3 bucket should not be publicly readable.

Remediation

To remediate this issue, update your S3 bucket to remove public access.

To remove public access from an S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the left navigation pane, choose **Buckets**.
3. Choose the name of the S3 bucket to update.
4. Choose **Permissions** and then choose **Block public access**.
5. Choose **Edit**.
6. Select **Block all public access**. Then choose **Save**.
7. If prompted, enter **confirm** and then choose **Confirm**.

[S3.3] S3 buckets should prohibit public write access

Category: Protect > Secure network configuration

Severity: Critical

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-bucket-public-write-prohibited](#)

Schedule type: Periodic and change triggered

Parameters: None

This control checks whether your S3 buckets allow public write access. It evaluates the block public access settings, the bucket policy, and the bucket access control list (ACL).

Some use cases require that everyone on the internet be able to write to your S3 bucket. However, those situations are rare. To ensure the integrity and security of your data, your S3 bucket should not be publicly writable.

Remediation

To remediate this issue, update your S3 bucket to remove public access.

To remove public access for an S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the left navigation pane, choose **Buckets**.
3. Choose the name of the S3 bucket to update.
4. Choose **Permissions** and then choose **Block public access**.
5. Choose **Edit**.
6. Select **Block all public access**. Then choose **Save**.
7. If prompted, enter **confirm** and then choose **Confirm**.

[S3.4] S3 buckets should have server-side encryption enabled

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS : : S3 : : Bucket

AWS Config rule: [s3-bucket-server-side-encryption-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks that your S3 bucket either has Amazon S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server-side encryption.

For an added layer of security for your sensitive data in S3 buckets, you should configure your buckets with server-side encryption to protect your data at rest. Amazon S3 encrypts each object with a unique key. As an additional safeguard, Amazon S3 encrypts the key itself with a root key that it rotates regularly. Amazon S3 server-side encryption uses one of the strongest block ciphers available to encrypt your data, 256-bit Advanced Encryption Standard (AES-256).

To learn more, see [Protecting data using server-side encryption with Amazon S3-managed encryption keys \(SSE-S3\)](#) in the *Amazon Simple Storage Service User Guide*.

Remediation

To remediate this issue, update your S3 bucket to enable default encryption.

To enable default encryption on an S3 bucket

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the left navigation pane, choose **Buckets**.
3. Choose the S3 bucket from the list.
4. Choose **Properties**.
5. Choose **Default encryption**.
6. For the encryption, choose either **AES-256** or **AWS-KMS**.

- Choose **AES-256** to use keys that are managed by Amazon S3 for default encryption. For more information about using Amazon S3 server-side encryption to encrypt your data, see the [Amazon Simple Storage Service User Guide](#).
- Choose **AWS-KMS** to use keys that are managed by AWS KMS for default encryption. Then choose a root key from the list of the AWS KMS root keys that you have created.

Type the Amazon Resource Name (ARN) of the AWS KMS key to use. You can find the ARN for your AWS KMS key in the IAM console, under **Encryption keys**. Or, you can choose a key name from the drop-down list.

Important

If you use the AWS KMS option for your default encryption configuration, you are subject to the RPS (requests per second) quotas of AWS KMS. For more information about AWS KMS quotas and how to request a quota increase, see the [AWS Key Management Service Developer Guide](#).

For more information about creating an AWS KMS key, see the [AWS Key Management Service Developer Guide](#).

For more information about using AWS KMS with Amazon S3, see the [Amazon Simple Storage Service User Guide](#).

When enabling default encryption, you might need to update your bucket policy. For more information about moving from bucket policies to default encryption, see the [Amazon Simple Storage Service User Guide](#).

7. Choose **Save**.

For more information about default S3 bucket encryption, see the [Amazon Simple Storage Service User Guide](#).

[S3.5] S3 buckets should require requests to use Secure Socket Layer

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS::S3::Bucket

AWS Config rule: `s3-bucket-ssl-requests-only`

Schedule type: Change triggered

Parameters: None

This control checks whether S3 buckets have policies that require requests to use Secure Socket Layer (SSL).

S3 buckets should have policies that require all requests (`Action: S3:*`) to only accept transmission of data over HTTPS in the S3 resource policy, indicated by the condition key `aws:SecureTransport`.

Remediation

To remediate this issue, update the permissions policy of the S3 bucket.

To configure an S3 bucket to deny nonsecure transport

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Navigate to the noncompliant bucket, then choose the bucket name.

3. Choose **Permissions**, and then choose **Bucket Policy**.
4. Add a similar policy statement to that in the policy below. Replace `awsexamplebucket` with the name of the bucket you are modifying.

```
{
  "Id": "ExamplePolicy",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSSLRequestsOnly",
      "Action": "s3:*",
      "Effect": "Deny",
      "Resource": [
        "arn:aws:s3:::awsexamplebucket",
        "arn:aws:s3:::awsexamplebucket/*"
      ],
      "Condition": {
        "Bool": {
          "aws:SecureTransport": "false"
        }
      },
      "Principal": "*"
    }
  ]
}
```

5. Choose **Save**.

For more information, see the knowledge center article [What S3 bucket policy should I use to comply with the AWS Config rule s3-bucket-ssl-requests-only?](#)

[S3.6] Amazon S3 permissions granted to other AWS accounts in bucket policies should be restricted

Category: Protect > Secure access management > Sensitive API operations actions restricted

Severity: High

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-bucket-blacklisted-actions-prohibited](#)

Schedule type: Change triggered

Parameters:

- `blacklistedactionpatterns`: `s3:DeleteBucketPolicy`, `s3:PutBucketAcl`, `s3:PutBucketPolicy`, `s3:PutEncryptionConfiguration`, `s3:PutObjectAcl`

This control checks whether the S3 bucket policy prevents principals from other AWS accounts from performing denied actions on resources in the S3 bucket. The control fails if the S3 bucket policy allows any of the following actions for a principal in another AWS account:

- `s3:DeleteBucketPolicy`
- `s3:PutBucketAcl`
- `s3:PutBucketPolicy`
- `s3:PutEncryptionConfiguration`
- `s3:PutObjectAcl`

Implementing least privilege access is fundamental to reducing security risk and the impact of errors or malicious intent. If an S3 bucket policy allows access from external accounts, it could result in data exfiltration by an insider threat or an attacker.

The `blacklistedactionpatterns` parameter allows for successful evaluation of the rule for S3 buckets. The parameter grants access to external accounts for action patterns that are not included in the `blacklistedactionpatterns` list.

Remediation

To remediate this issue, edit the S3 bucket policy to remove the permissions.

To edit an S3 bucket policy

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the **Bucket name** list, choose the name of the S3 bucket for which you want to edit the policy.
3. Choose **Permissions**, and then choose **Bucket Policy**.
4. In the **Bucket policy editor** text box, do one of the following:
 - Remove the statements that grant access to denied actions to other AWS accounts
 - Remove the permitted denied actions from the statements
5. Choose **Save**.

[S3.8] S3 Block Public Access setting should be enabled at the bucket level

Category: Protect > Secure access management > Access control

Severity: High

Resource type: AWS::S3::Bucket

AWS Config rule: `s3-bucket-level-public-access-prohibited`

Schedule type: Change triggered

Parameters:

- `excludedPublicBuckets` (Optional) – A comma-separated list of known allowed public S3 bucket names.

This control checks whether S3 buckets have bucket-level public access blocks applied. This control fails if any of the following settings are set to `false`:

- `ignorePublicAcls`
- `blockPublicPolicy`
- `blockPublicAcls`
- `restrictPublicBuckets`

Block Public Access at the S3 bucket level provides controls to ensure that objects never have public access. Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, or both.

Unless you intend to have your S3 buckets publicly accessible, you should configure the bucket level Amazon S3 Block Public Access feature.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information on how to remove public access at a bucket level, see [Blocking public access to your Amazon S3 storage](#) in the *Amazon S3 User Guide*.

[S3.9] S3 bucket server access logging should be enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS : : S3 : : Bucket

AWS Config rule: [s3-bucket-logging-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether server access logging is enabled for S3 buckets. When logging is enabled, Amazon S3 delivers access logs for a source bucket to a chosen target bucket. The target bucket must be in the same AWS Region as the source bucket and must not have a default retention period configuration. This control passes if server access logging is enabled. The target logging bucket does not need to have server access logging enabled, and you should suppress findings for this bucket.

Server access logging provides detailed records of requests made to a bucket. Server access logs can assist in security and access audits. For more information, see [Security Best Practices for Amazon S3: Enable Amazon S3 server access logging](#).

Remediation

To enable S3 bucket access logging

1. Sign in to the AWS Management Console and open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Select the bucket from the list.
3. Choose **Properties**.
4. Under **Server access logging**, choose **Edit**.
5. Under **Server access logging**, choose **Enable**. Then, choose **Save changes**.

[S3.10] S3 buckets with versioning enabled should have lifecycle policies configured

Category: Identify > Logging

Severity: Medium

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-version-lifecycle-policy-check](#)

Schedule type: Change triggered

Parameters: None

This control checks if Amazon Simple Storage Service (Amazon S3) version enabled buckets have lifecycle policy configured. This rule fails if Amazon S3 lifecycle policy is not enabled.

It is recommended to configure lifecycle rules on your Amazon S3 bucket as these rules help you define actions that you want Amazon S3 to take during an object's lifetime.

Remediation

For more information on configuring lifecycle on an Amazon S3 bucket, see [Setting lifecycle configuration on a bucket](#) and [Managing your storage lifecycle](#).

[S3.11] S3 buckets should have event notifications enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-event-notifications-enabled](#)

Schedule type: Change triggered

Parameters: None

This control checks whether S3 Event Notifications are enabled on an Amazon S3 bucket. This control fails if S3 Event Notifications are not enabled on a bucket.

By enabling Event Notifications, you receive alerts on your Amazon S3 buckets when specific events occur. For example, you can be notified of object creation, object removal, and object restoration. These notifications can alert relevant teams to accidental or intentional modifications that may lead to unauthorized data access.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information about detecting changes to S3 buckets and objects, see [Amazon S3 Event Notifications](#) in the *Amazon S3 User Guide*.

[S3.12] S3 access control lists (ACLs) should not be used to manage user access to buckets

Category: Protect > Secure access management > Access control

Severity: Medium

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-bucket-acl-prohibited](#)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon S3 buckets provide user permissions via ACLs. The control fails if ACLs are configured for managing user access on S3 buckets.

ACLs are legacy access control mechanisms that predate IAM. Instead of ACLs, we recommend using IAM policies or S3 bucket policies to more easily manage access to your S3 buckets.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For more information on managing access to S3 buckets, see [Bucket policies and user policies](#) in the *Amazon S3 User Guide*. For details on how to review your current ACL permissions, see [Access control list \(ACL\) overview](#) in the *Amazon S3 User Guide*.

[S3.13] S3 buckets should have lifecycle policies configured

Category: Protect > Data protection

Severity: Low

Resource type: AWS::S3::Bucket

AWS Config rule: [s3-lifecycle-policy-check](#)

Schedule type: Change triggered

Parameters: None

This control checks if a lifecycle policy is configured for an Amazon S3 bucket. This control fails if a lifecycle policy is not configured for an S3 bucket.

Configuring lifecycle rules on your S3 bucket defines actions that you want S3 to take during an object's lifetime. For example, you can transition objects to another storage class, archive them, or delete them after a specified period of time.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)

- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For information about configuring lifecycle policies on an Amazon S3 bucket, see [Setting lifecycle configuration on a bucket](#) and see [Managing your storage lifecycle](#) in the *Amazon S3 User Guide*.

[SageMaker.1] SageMaker notebook instances should not have direct internet access

Category: Protect > Secure network configuration

Severity: High

Resource type: AWS::SageMaker::NotebookInstance

AWS Config rule: [sagemaker-notebook-no-direct-internet-access](#)

Schedule type: Periodic

Parameters: None

This control checks whether direct internet access is disabled for an SageMaker notebook instance. To do this, it checks whether the `DirectInternetAccess` field is disabled for the notebook instance.

If you configure your SageMaker instance without a VPC, then by default direct internet access is enabled on your instance. You should configure your instance with a VPC and change the default setting to **Disable — Access the internet through a VPC**.

To train or host models from a notebook, you need internet access. To enable internet access, make sure that your VPC has a NAT gateway and your security group allows outbound connections. To learn more about how to connect a notebook instance to resources in a VPC, see [Connect a notebook instance to resources in a VPC](#) in the *Amazon SageMaker Developer Guide*.

You should also ensure that access to your SageMaker configuration is limited to only authorized users. Restrict users' IAM permissions to modify SageMaker settings and resources.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- China (Beijing)
- China (Ningxia)
- Europe (Milan)
- AWS GovCloud (US-East)

Remediation

Note that you cannot change the internet access setting after a notebook instance is created. It must be stopped, deleted, and recreated.

To configure an SageMaker notebook instance to deny direct internet access

1. Open the SageMaker console at <https://console.aws.amazon.com/sagemaker/>
2. Navigate to **Notebook instances**.

3. Delete the instance that has direct internet access enabled. Choose the instance, choose **Actions**, then choose stop.

After the instance is stopped, choose **Actions**, then choose **delete**.

4. Choose **Create notebook instance**. Provide the configuration details.
5. Expand the network section, then choose a VPC, subnet, and security group. Under **Direct internet access**, choose **Disable — Access the internet through a VPC**.
6. Choose **Create notebook instance**.

For more information, see [Connect a notebook instance to resources in a VPC](#) in the *Amazon SageMaker Developer Guide*.

[SecretsManager.1] Secrets Manager secrets should have automatic rotation enabled

Category: Protect > Secure development

Severity: Medium

Resource type: AWS::SecretsManager::Secret

AWS Config rule: [secretsmanager-rotation-enabled-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether a secret stored in AWS Secrets Manager is configured with automatic rotation.

Secrets Manager helps you improve the security posture of your organization. Secrets include database credentials, passwords, and third-party API keys. You can use Secrets Manager to store secrets centrally, encrypt secrets automatically, control access to secrets, and rotate secrets safely and automatically.

Secrets Manager can rotate secrets. You can use rotation to replace long-term secrets with short-term ones. Rotating your secrets limits how long an unauthorized user can use a compromised secret. For this reason, you should rotate your secrets frequently. To learn more about rotation, see [Rotating your AWS Secrets Manager secrets](#) in the *AWS Secrets Manager User Guide*.

Remediation

To remediate this issue, you enable automatic rotation for your secrets.

To enable automatic rotation for secrets

1. Open the Secrets Manager console at <https://console.aws.amazon.com/secretsmanager/>.
2. To find the secret that requires rotating, enter the secret name in the search field.
3. Choose the secret you want to rotate, which displays the secrets details page.
4. Under **Rotation configuration**, choose **Edit rotation**.
5. From **Edit rotation configuration**, choose **Enable automatic rotation**.
6. For **Select Rotation Interval**, choose a rotation interval.
7. Choose a Lambda function for rotation. For information about customizing your Lambda rotation function, see [Understanding and customizing your Lambda rotation function](#) in the *AWS Secrets Manager User Guide*.
8. To configure the secret for rotation, choose **Next**.

To learn more about Secrets Manager rotation, see [Rotating your AWS Secrets Manager secrets](#) in the *AWS Secrets Manager User Guide*.

[SecretsManager.2] Secrets Manager secrets configured with automatic rotation should rotate successfully

Category: Protect > Secure development

Severity: Medium

Resource type: AWS::SecretsManager::Secret

AWS Config rule: [secretsmanager-scheduled-rotation-success-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS Secrets Manager secret rotated successfully based on the rotation schedule. The control fails if `RotationOccurringAsScheduled` is `false`. The control does not evaluate secrets that do not have rotation configured.

Secrets Manager helps you improve the security posture of your organization. Secrets include database credentials, passwords, and third-party API keys. You can use Secrets Manager to store secrets centrally, encrypt secrets automatically, control access to secrets, and rotate secrets safely and automatically.

Secrets Manager can rotate secrets. You can use rotation to replace long-term secrets with short-term ones. Rotating your secrets limits how long an unauthorized user can use a compromised secret. For this reason, you should rotate your secrets frequently.

In addition to configuring secrets to rotate automatically, you should ensure that those secrets rotate successfully based on the rotation schedule.

To learn more about rotation, see [Rotating your AWS Secrets Manager secrets](#) in the *AWS Secrets Manager User Guide*.

Remediation

If the automatic rotation fails, then Secrets Manager might have encountered errors with the configuration.

To rotate secrets in Secrets Manager, you use a Lambda function that defines how to interact with the database or service that owns the secret.

For help on how to diagnose and fix common errors related to secrets rotation, see [Troubleshooting AWS Secrets Manager rotation of secrets](#) in the *AWS Secrets Manager User Guide*.

[SecretsManager.3] Remove unused Secrets Manager secrets

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS::SecretsManager::Secret

AWS Config rule: [secretsmanager-secret-unused](#)

Schedule type: Periodic

Parameters: None

This control checks whether your secrets have been accessed within a specified number of days. The default value is 90 days. If a secret was not accessed within the defined number of days, this control fails.

Deleting unused secrets is as important as rotating secrets. Unused secrets can be abused by their former users, who no longer need access to these secrets. Also, as more users get access to a secret, someone might have mishandled and leaked it to an unauthorized entity, which increases the risk of abuse. Deleting unused secrets helps revoke secret access from users who no longer need it. It also helps to reduce the cost of using Secrets Manager. Therefore, it is essential to routinely delete unused secrets.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can delete inactive secrets from the Secrets Manager console.

To delete inactive secrets

1. Open the Secrets Manager console at <https://console.aws.amazon.com/secretsmanager/>.
2. To locate the secret, enter the secret name in the search box.
3. Choose the secret to delete.
4. Under **Secret details**, from **Actions**, choose **Delete secret**.
5. Under **Schedule secret deletion**, enter the number of days to wait before the secret is deleted.
6. Choose **Schedule deletion**.

[SecretsManager.4] Secrets Manager secrets should be rotated within a specified number of days

Category: Protect > Secure access management

Severity: Medium

Resource type: AWS::SecretsManager::Secret

AWS Config rule: [secretsmanager-secret-periodic-rotation](#)

Schedule type: Periodic

Parameters:

- **Rotation period:** 90 days by default

This control checks whether your secrets have been rotated at least once within 90 days.

Rotating secrets can help you to reduce the risk of an unauthorized use of your secrets in your AWS account. Examples include database credentials, passwords, third-party API keys, and even arbitrary text. If you do not change your secrets for a long period of time, the secrets are more likely to be compromised.

As more users get access to a secret, it can become more likely that someone mishandled and leaked it to an unauthorized entity. Secrets can be leaked through logs and cache data. They can be shared for debugging purposes and not changed or revoked once the debugging completes. For all these reasons, secrets should be rotated frequently.

You can configure your secrets for automatic rotation in AWS Secrets Manager. With automatic rotation, you can replace long-term secrets with short-term ones, significantly reducing the risk of compromise.

Security Hub recommends that you enable rotation for your Secrets Manager secrets. To learn more about rotation, see [Rotating your AWS Secrets Manager secrets](#) in the *AWS Secrets Manager User Guide*.

Note

This control is not supported in the following Regions:

- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can enable automatic secret rotation in the Secrets Manager console.

To enable secret rotation

1. Open the Secrets Manager console at <https://console.aws.amazon.com/secretsmanager/>.
2. To locate the secret, enter the secret name in the search box.
3. Choose the secret to display.
4. Under **Rotation configuration**, choose **Edit rotation**.
5. From **Edit rotation configuration**, choose **Enable automatic rotation**.
6. From **Select Rotation Interval**, choose the rotation interval.
7. Choose a Lambda function to use for rotation.
8. Choose **Next**.
9. After you configure the secret for automatic rotation, under **Rotation Configuration**, choose **Rotate secret immediately**.

[SNS.1] SNS topics should be encrypted at rest using AWS KMS

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::SNS::Topic

AWS Config rule: [sns-encrypted-kms](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an SNS topic is encrypted at rest using AWS KMS.

Encrypting data at rest reduces the risk of data stored on disk being accessed by a user not authenticated to AWS. It also adds another set of access controls to limit the ability of unauthorized users to access the

data. For example, API permissions are required to decrypt the data before it can be read. SNS topics should be encrypted at-rest for an added layer of security. For more information, see [Encryption at rest](#) in the *Amazon Simple Notification Service Developer Guide*.

Remediation

To remediate this issue, update your SNS topic to enable encryption.

To encrypt an unencrypted SNS topic

1. Open the Amazon SNS console at <https://console.aws.amazon.com/sns/v3/home>.
2. In the navigation pane, choose **Topics**.
3. Choose the name of the topic to encrypt.
4. Choose **Edit**.
5. Under **Encryption**, choose **Enable Encryption**.
6. Choose the KMS key to use to encrypt the topic.
7. Choose **Save changes**.

[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic

Category: Identify > Logging

Severity: Medium

Resource type: AWS::SNS::Topic

AWS Config rule: `sns-topic-message-delivery-notification-enabled`

Schedule type: Change triggered

Parameters: None

This control checks whether logging is enabled for the delivery status of notification messages sent to an Amazon SNS topic for the endpoints. This control fails if the delivery status notification for messages is not enabled.

Logging is an important part of maintaining the reliability, availability, and performance of services. Logging message delivery status helps provide operational insights, such as the following:

- Knowing whether a message was delivered to the Amazon SNS endpoint.
- Identifying the response sent from the Amazon SNS endpoint to Amazon SNS.
- Determining the message dwell time (the time between the publish timestamp and the hand off to an Amazon SNS endpoint).

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)

- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To configure delivery status logging for a topic, see [Amazon SNS message delivery status](#) in the *Amazon Simple Notification Service Developer Guide*.

[SQS.1] Amazon SQS queues should be encrypted at rest

Category: Protect > Data protection > Encryption of data at rest

Severity: Medium

Resource type: AWS::SQS::Queue

AWS Config rule: sqs-queue-encrypted (Custom rule developed by Security Hub)

Schedule type: Change triggered

Parameters: None

This control checks whether Amazon SQS queues are encrypted at rest. The control passes if you use an Amazon SQS managed key (SSE-SQS) or an AWS Key Management Service (AWS KMS) key (SSE-KMS).

Server-side encryption (SSE) allows you to transmit sensitive data in encrypted queues. To protect the content of messages in queues, SSE uses KMS keys. For more information, see [Encryption at rest](#) in the *Amazon Simple Queue Service Developer Guide*.

Remediation

For information about managing SSE using the AWS Management Console, see [Configuring server-side encryption \(SSE\) for a queue \(console\)](#) in the *Amazon Simple Queue Service Developer Guide*.

[SSM.1] EC2 instances should be managed by AWS Systems Manager

Category: Identify > Inventory

Severity: Medium

Resource type: AWS::EC2::Instance

AWS Config rule: ec2-instance-managed-by-systems-manager

Schedule type: Change triggered

Parameters: None

This control checks whether the stopped and running EC2 instances in your account are managed by AWS Systems Manager. Systems Manager is an AWS service that you can use to view and control your AWS infrastructure.

To help you to maintain security and compliance, Systems Manager scans your stopped and running managed instances. A managed instance is a machine that is configured for use with Systems Manager. Systems Manager then reports or takes corrective action on any policy violations that it detects. Systems Manager also helps you to configure and maintain your managed instances.

To learn more, see [AWS Systems Manager User Guide](#).

Remediation

You can use the Systems Manager console to remediate this issue.

To ensure that EC2 instances are managed by Systems Manager

1. Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>.
2. In the navigation menu, choose **Quick setup**.
3. Choose **Create**.
4. Under **Configuration type**, choose **Host Management**, then choose **Next**.
5. On the configuration screen, you can keep the default options.

You can optionally make the following changes:

- a. If you use CloudWatch to monitor EC2 instances, select **Install and configure the CloudWatch agent** and **Update the CloudWatch agent once every 30 days**.
 - b. Under **Targets**, choose the management scope to determine the accounts and Regions where this configuration is applied.
 - c. Under **Instance profile options**, select **Add required IAM policies to existing instance profiles attached to your instances**.
6. Choose **Create**.

To determine whether your instances support Systems Manager associations, see [Systems Manager prerequisites](#) in the *AWS Systems Manager User Guide*.

[SSM.2] All EC2 instances managed by Systems Manager should be compliant with patching requirements

Category: Detect > Detection services

Severity: High

Resource type: AWS::SSM::PatchCompliance

AWS Config rule: [ec2-managedinstance-patch-compliance-status-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the compliance status of the Amazon EC2 Systems Manager patch compliance is `COMPLIANT` or `NON_COMPLIANT` after the patch installation on the instance. It only checks instances that are managed by Systems Manager Patch Manager.

Having your EC2 instances fully patched as required by your organization reduces the attack surface of your AWS accounts.

Note

This control is not supported in the following Regions:

- Africa (Cape Town)
- Europe (Milan)
- Middle East (Bahrain)

Remediation

To remediate this issue, install the required patches on your noncompliant instances.

To remediate noncompliant patches

1. Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>.
2. Under **Node Management**, choose **Run Command** and then choose **Run command**.
3. Choose the button next to **AWS-RunPatchBaseline**.
4. Change the **Operation** to **Install**.
5. Choose **Choose instances manually** and then choose the noncompliant instances.
6. At the bottom of the page, choose **Run**.
7. After the command is complete, to monitor the new compliance status of your patched instances, in the navigation pane, choose **Compliance**.

For more information about using Systems Manager documents to patch a managed instance, see [About SSM documents for patching instances](#) and [Running commands using Systems Manager Run command](#) in the *AWS Systems Manager User Guide*.

[SSM.3] Instances managed by Systems Manager should have an association compliance status of COMPLIANT

Category: Detect > Detection services

Severity: Low

Resource type: AWS::SSM::AssociationCompliance

AWS Config rule: [ec2-managedinstance-association-compliance-status-check](#)

Schedule type: Change triggered

Parameters: None

This control checks whether the status of the AWS Systems Manager association compliance is `COMPLIANT` or `NON_COMPLIANT` after the association is run on an instance. The control passes if the association compliance status is `COMPLIANT`.

A State Manager association is a configuration that is assigned to your managed instances. The configuration defines the state that you want to maintain on your instances. For example, an association can specify that antivirus software must be installed and running on your instances or that certain ports must be closed.

After you create one or more State Manager associations, compliance status information is immediately available to you. You can view the compliance status in the console or in response to AWS CLI commands or corresponding Systems Manager API actions. For associations, Configuration Compliance shows the compliance status (`Compliant` or `Non-compliant`). It also shows the severity level assigned to the association, such as `Critical` or `Medium`.

To learn more about State Manager association compliance, see [About State Manager association compliance](#) in the *AWS Systems Manager User Guide*.

Note

This control is not supported in Africa (Cape Town) or Europe (Milan).

Remediation

A failed association can be related to different things, including targets and SSM document names. To remediate this issue, you must first identify and investigate the association. You can then update the association to correct the specific issue.

You can edit an association to specify a new name, schedule, severity level, or targets. After you edit an association, AWS Systems Manager creates a new version.

To investigate and update a failed association

1. Open the AWS Systems Manager console at <https://console.aws.amazon.com/systems-manager/>.
2. In the navigation pane, under **Node Management**, choose **Fleet Manager**.
3. Choose the instance ID that has an **Association status** of **Failed**.
4. Choose **View details**.
5. Choose **Associations**.
6. Note the name of the association that has an **Association status** of **Failed**. This is the association that you need to investigate. You need to use the association name in the next step.
7. In the navigation pane, under **Node Management**, choose **State Manager**. Search for the association name, then select the association.
8. After you determine the issue, edit the failed association to correct the problem. For information on how to edit an association, see [Edit an association](#).

For more information on creating and editing State Manager associations, see [Working with associations in Systems Manager](#) in the *AWS Systems Manager User Guide*.

[SSM.4] SSM documents should not be public

Category: Protect > Secure network configuration > Resources not publicly accessible

Severity: Critical

Resource type: AWS :: SSM :: Document

AWS Config rule: `ssm-document-not-public`

Schedule type: Periodic

Parameters: None

This control checks whether AWS Systems Manager documents that are owned by the account are public. This control fails if SSM documents with the owner `self` are public.

SSM documents that are public might allow unintended access to your documents. A public SSM document can expose valuable information about your account, resources, and internal processes.

Unless your use case requires public sharing to be enabled, Security Hub recommends that you turn on the block public sharing setting for your Systems Manager documents that are owned by `self`.

Note

This control is not supported in the following Regions:

- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

For more information about disabling public access to SSM documents, see [Modify permissions for a shared SSM document](#) and [Best practices for shared SSM documents](#) in the *AWS Systems Manager User Guide*.

[WAF.1] AWS WAF Classic global web ACL logging should be enabled

Category: Identify > Logging

Severity: Medium

Resource type: AWS : : WAF : : WebACL

AWS Config rule: [waf-classic-logging-enabled](#)

Schedule type: Periodic

Parameters: None

This control checks whether logging is enabled for an AWS WAF global web ACL. This control fails if logging is not enabled for the web ACL.

Logging is an important part of maintaining the reliability, availability, and performance of AWS WAF globally. It is a business and compliance requirement in many organizations, and allows you to troubleshoot application behavior. It also provides detailed information about the traffic that is analyzed by the web ACL that is attached to AWS WAF.

Note

This control is not supported in the following Regions:

- US East (Ohio)
- US West (N. California)
- US West (Oregon)
- Africa (Cape Town)
- Asia Pacific (Hong Kong)
- Asia Pacific (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seoul)
- Asia Pacific (Singapore)
- Asia Pacific (Sydney)
- Asia Pacific (Tokyo)
- Canada (Central)
- China (Beijing)
- China (Ningxia)
- Europe (Frankfurt)
- Europe (Ireland)
- Europe (London)
- Europe (Milan)
- Europe (Paris)
- Europe (Stockholm)
- Middle East (Bahrain)
- South America (São Paulo)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

You can enable logging for a web ACL from the Kinesis Data Firehose console.

To enable logging for a web ACL

1. Open the Kinesis Data Firehose console at <https://console.aws.amazon.com/firehose/>.
2. Create a Kinesis Data Firehose delivery stream.

The name must start with the prefix `aws-waf-logs-`. For example, `aws-waf-logs-us-east-2-analytics`.

Create the Kinesis Data Firehose delivery stream with a `PUT` source and in the Region where you operate. If you capture logs for Amazon CloudFront, create the delivery stream in US East (N. Virginia). For more information, see [Creating an Amazon Kinesis Data Firehose delivery stream](#) in the *Amazon Kinesis Data Firehose Developer Guide*.

3. From **Services**, choose **WAF & Shield**. Then choose **Switch to AWS WAF Classic**.
4. From **Filter**, choose **Global (CloudFront)**.
5. Choose the web ACL to enable logging for.
6. Under **Logging**, choose **Enable logging**.
7. Choose the Kinesis Data Firehose delivery stream that you created earlier. You must choose a delivery stream that has a name that begins with `aws-waf-logs-`.
8. Choose **Enable logging**.

[WAF.2] A WAF Regional rule should have at least one condition

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFRegional::Rule`

AWS Config rule: `waf-regional-rule-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF Regional rule has at least one condition. The control fails if no conditions are present within a rule.

A WAF Regional rule can contain multiple conditions. The rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any conditions, the traffic passes without inspection. A WAF Regional rule with no conditions, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To add a condition to an empty rule, see [Adding and removing conditions in a rule](#) in the *AWS WAF Developer Guide*.

[WAF.3] A WAF Regional rule group should have at least one rule

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFRegional::RuleGroup`

AWS Config rule: `waf-regional-rulegroup-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF Regional rule group has at least one rule. The control fails if no rules are present within a rule group.

A WAF Regional rule group can contain multiple rules. The rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any rules, the traffic passes without inspection. A WAF Regional rule group with no rules, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- Asia Pacific (Osaka)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To add rules and rule conditions to an empty rule group, see [Adding and deleting rules from an AWS WAF Classic rule group](#) and [Adding and removing conditions in a rule](#) in the *AWS WAF Developer Guide*.

[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group

Category: Protect > Secure network configuration

Severity: Medium

Resource type: `AWS::WAFRegional::WebACL`

AWS Config rule: `waf-regional-webacl-not-empty`

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF Classic Regional web ACL contains any WAF rules or WAF rule groups. This control fails if a web ACL does not contain any WAF rules or rule groups.

A WAF Regional web ACL can contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by WAF depending on the default action.

Note

This control is not supported in the following Regions:

- Asia Pacific (Jakarta)
- China (Beijing)
- China (Ningxia)
- AWS GovCloud (US-East)
- AWS GovCloud (US-West)

Remediation

To add rules or rule groups to an empty web ACL

1. Open the AWS WAF console at <https://console.aws.amazon.com/wafv2/>.
2. In the navigation pane, choose **Switch to AWS WAF Classic**, and then choose **Web ACLs**.
3. For **Filter**, choose the Region where the empty web ACL is located.
4. Choose the name of the empty web ACL.
5. Choose **Rules**, and then choose **Edit web ACL**.
6. For **Rules**, choose a rule or rule group, and then choose **Add rule to web ACL**.
7. At this point, you can modify the rule order within the web ACL if you are adding multiple rules or rule groups to the web ACL.
8. Choose **Update**.

[WAF.6] A WAF global rule should have at least one condition

Category: Protect > Secure network configuration

Severity: Medium

Resource type: AWS::WAF::Rule

AWS Config rule: [waf-global-rule-not-empty](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF global rule contains any conditions. The control fails if no conditions are present within a rule.

A WAF global rule can contain multiple conditions. A rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any conditions, the traffic passes without inspection. A WAF global rule with no conditions, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Note

This control is only supported in US East (N. Virginia).

Remediation

For instructions on creating a rule and adding conditions, see [Creating a rule and adding conditions](#) in the *AWS WAF Developer Guide*.

[WAF.7] A WAF global rule group should have at least one rule

Category: Protect > Secure network configuration

Severity: Medium

Resource type: AWS::WAF::RuleGroup

AWS Config rule: [waf-global-rulegroup-not-empty](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF global rule group has at least one rule. The control fails if no rules are present within a rule group.

A WAF global rule group can contain multiple rules. The rule's conditions allow for traffic inspection and take a defined action (allow, block, or count). Without any rules, the traffic passes without inspection. A WAF global rule group with no rules, but with a name or tag suggesting allow, block, or count, could lead to the wrong assumption that one of those actions is occurring.

Note

This control is only supported in US East (N. Virginia).

Remediation

For instructions on adding a rule to a rule group, see [Creating an AWS WAF Classic rule group](#) in the *AWS WAF Developer Guide*.

[WAF.8] A WAF global web ACL should have at least one rule or rule group

Category: Protect > Secure network configuration

Severity: Medium

Resource type: AWS::WAF::WebACL

AWS Config rule: [waf-global-webacl-not-empty](#)

Schedule type: Change triggered

Parameters: None

This control checks whether an AWS WAF global web ACL contains at least one WAF rule or WAF rule group. The control fails if a web ACL does not contain any WAF rules or rule groups.

A WAF global web ACL can contain a collection of rules and rule groups that inspect and control web requests. If a web ACL is empty, the web traffic can pass without being detected or acted upon by WAF depending on the default action.

Note

This control is only supported in US East (N. Virginia).

Remediation

To add rules or rule groups to an empty web ACL

1. Open the AWS WAF console at <https://console.aws.amazon.com/wafv2/>.
2. In the navigation pane, choose **Switch to AWS WAF Classic**, and then choose **Web ACLs**.
3. For **Filter**, choose **Global (CloudFront)**.
4. Choose the name of the empty web ACL.
5. Choose **Rules**, and then choose **Edit web ACL**.

6. For **Rules**, choose a rule or rule group, and then choose **Add rule to web ACL**.
7. At this point, you can modify the rule order within the web ACL if you are adding multiple rules or rule groups to the web ACL.
8. Choose **Update**.

Control categories

Each control in AWS Security Hub is assigned a category. The category for a control reflects the security function that the control applies to.

The category value contains the category, the subcategory within the category, and, optionally, a classifier within the subcategory. For example:

- Detect > Detection services > Application monitoring
- Identify > Inventory
- Protect > Data protection > Encryption of data in transit

Here are descriptions of the categories, subcategories, and classifiers that apply to currently available Security Hub controls.

Detect

Develop and implement the appropriate activities to identify the occurrence of a cybersecurity event.

Detection services

Are the correct detection services in place with the right amount of coverage?

Examples of AWS detection services include Amazon CloudWatch Alarms, Amazon Detective, Amazon GuardDuty, Amazon Inspector, AWS IoT Device Defender, AWS Security Hub, and AWS Trusted Advisor.

- **Application monitoring** – Is application health monitored to maintain availability?

Identify

Develop the organizational understanding to manage cybersecurity risk to systems, assets, data, and capabilities.

Inventory

What resources are being used, and are they approved resources for this service?

- **Tagging** – Are the correct resource tagging strategies implemented for the service, including the resource owner?

Logging

Have you securely enabled all relevant logging for the service? Examples of log files include the following:

- Amazon VPC Flow Logs
- Elastic Load Balancing access logs
- Amazon CloudFront logs
- Amazon CloudWatch logs
- Amazon Relational Database Service logs
- Amazon OpenSearch Service slow index logs

- X-Ray tracing
- AWS Directory Service logs
- AWS Config items

Resource configuration

Do you understand how your resources are configured so that you can reduce your attack surface?

Vulnerability, patch, and version management

Do you have resources that need to be patched or resources with unacceptable vulnerabilities? Are you using the latest versions of software?

Protect

Develop and implement the appropriate safeguards to ensure delivery of critical infrastructure services and secure coding practices.

Secure access management

Are there strong authentication and authorization policies in place for the service?

- **Access control** – Do IAM or resource policies for the service align to least privilege practices?
- **Passwordless authentication** – Are federated identities or SSO used to authenticate users instead of IDs, passwords, and access keys? Is Kerberos used to eliminate the need to transmit passwords over the network?
- **Root user access restrictions** – Is usage of the root user being avoided?
- **Sensitive API actions restricted** – Are sensitive API actions for a service properly restricted, especially those leading to privilege escalation or resource-sharing? This could apply to IAM or resource-based policies.

Secure network configuration

Are public and insecure remote network access avoided for the service?

- **API private access** – Are VPC endpoints enabled for private access to AWS APIs?
- **Resources not publicly accessible** – Are resources properly segmented and isolated?
- **Resources within VPC** – Is there proper usage of VPCs, such as requiring jobs to run in VPCs?
- **Security group configuration** – Are security groups securely configured?

Data protection

Do you have mechanisms to protect data that your service consumes, sends, or stores?

- **Encryption of data at rest** – Does the service encrypt data at rest?
- **Encryption of data in transit** – Does the service encrypt data in transit?
- **Data integrity** – Does the service validate data for integrity?
- **Data deletion protection** – Does the service protect data from accidental deletion?

API protection

Does the service use AWS PrivateLink to protect the service API operations?

Protective services

Are the correct protective services in place? Do they provide the correct amount of coverage?

Protective services help you deflect attacks and compromises that are directed at the service. Examples of protective services in AWS include AWS Control Tower, AWS WAF, AWS Shield Advanced, AWS Network Firewall, AWS Secrets Manager, AWS Identity and Access Management Access Analyzer, and AWS Resource Access Manager.

Secure development

Do you use secure coding practices?

- **Credentials not hardcoded** – Do you have credentials hardcoded into your code?

Recover

Develop and implement the appropriate activities to maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity event.

Resilience

Can your workloads quickly recover from a security event?

- **Backups enabled** – Have you established and tested backups?
- **High availability** – Does the configuration of the service allow for graceful failovers, elastic scaling, and high availability?

AWS Foundational Best Practices controls that you might want to disable

To save on the cost of AWS Config, you can disable recording of global resources in all but one Region, and then disable these controls that deal with global resources in all Regions except for the Region that runs global recording.

- [\[IAM.1\] IAM policies should not allow full "*" administrative privileges \(p. 529\)](#)
- [\[IAM.2\] IAM users should not have IAM policies attached \(p. 529\)](#)
- [\[IAM.3\] IAM users' access keys should be rotated every 90 days or less \(p. 530\)](#)
- [\[IAM.4\] IAM root user access key should not exist \(p. 531\)](#)
- [\[IAM.5\] MFA should be enabled for all IAM users that have a console password \(p. 532\)](#)
- [\[IAM.6\] Hardware MFA should be enabled for the root user \(p. 533\)](#)
- [\[IAM.7\] Password policies for IAM users should have strong configurations \(p. 534\)](#)
- [\[IAM.8\] Unused IAM user credentials should be removed \(p. 535\)](#)

If you disable these controls and disable recording of global resources in a particular Region, you should also disable [\[Config.1\] AWS Config should be enabled \(p. 482\)](#). This is because [\[Config.1\] AWS Config should be enabled \(p. 482\)](#) requires recording of global resources in order to pass.

Logging AWS Security Hub API calls with AWS CloudTrail

AWS Security Hub is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Security Hub. CloudTrail captures API calls for Security Hub as events. The captured calls include calls from the Security Hub console and code calls to the Security Hub API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Security Hub. If you don't configure a trail, you can still view the most recent events on the CloudTrail console in **Event history**. Using the information that CloudTrail collects, you can determine the request that was made to Security Hub, the IP address that the request was made from, who made the request, when it was made, and additional details.

To learn more about CloudTrail, including how to configure and enable it, see the [AWS CloudTrail User Guide](#).

Security Hub information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When supported event activity occurs in Security Hub, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your account. For more information, see [Viewing events with CloudTrail event history](#).

For an ongoing record of events in your account, including events for Security Hub, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail on the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- [Overview for creating a trail](#)
- [CloudTrail supported services and integrations](#)
- [Configuring Amazon SNS notifications for CloudTrail](#)
- [Receiving CloudTrail log files from multiple regions](#) and [Receiving CloudTrail log files from multiple accounts](#)

Security Hub supports logging all of the Security Hub API actions as events in CloudTrail logs. To view a list of Security Hub operations, see the [Security Hub API Reference](#).

When activity for the following actions is logged to CloudTrail, the value for `responseElements` is set to `null`. This ensures that sensitive information isn't included in CloudTrail logs.

- `BatchImportFindings`
- `GetFindings`
- `GetInsights`
- `GetMembers`
- `UpdateFindings`

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or AWS Identity and Access Management (IAM) user credentials
- Whether the request was made with temporary security credentials for a role or federated user
- Whether the request was made by another AWS service

For more information, see the [CloudTrail userIdentity element](#).

Example: Security Hub log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateInsight` action. In this example, an insight called `Test Insight` is created. The `ResourceId` attribute is specified as the **Group by** aggregator, and no optional filters for this insight are specified. For more information about insights, see [Insights in AWS Security Hub \(p. 277\)](#).

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDAJK6U5DS22IAVUI7BW",
    "arn": "arn:aws:iam::012345678901:user/TestUser",
    "accountId": "012345678901",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "TestUser"
  },
  "eventTime": "2018-11-25T01:02:18Z",
  "eventSource": "securityhub.amazonaws.com",
  "eventName": "CreateInsight",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "205.251.233.179",
  "userAgent": "aws-cli/1.11.76 Python/2.7.10 Darwin/17.7.0 botocore/1.5.39",
  "requestParameters": {
    "Filters": {},
    "ResultField": "ResourceId",
    "Name": "Test Insight"
  },
  "responseElements": {
    "InsightArn": "arn:aws:securityhub:us-west-2:0123456789010:insight/custom/f4c4890b-ac6b-4c26-95f9-e62cc46f3055"
  },
  "requestID": "c0fffcdd-f04d-11e8-93fc-ddcd14710066",
  "eventID": "3dabcebf-35b0-443f-a1a2-26e186ce23bf",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "012345678901"
}
```

Automated response and remediation

With Amazon EventBridge, you can automate your AWS services to respond automatically to system events such as application availability issues or resource changes. Events from AWS services are delivered to EventBridge in near-real time and on a guaranteed basis. You can write simple rules to indicate which events you are interested in and what automated actions to take when an event matches a rule. The actions that can be automatically triggered include the following:

- Invoking an AWS Lambda function
- Invoking the Amazon EC2 run command
- Relaying the event to Amazon Kinesis Data Streams
- Activating an AWS Step Functions state machine
- Notifying an Amazon SNS topic or an Amazon SQS queue
- Sending a finding to a third-party ticketing, chat, SIEM, or incident response and management tool

Security Hub automatically sends all new findings and all updates to existing findings to EventBridge as EventBridge events. You can also create custom actions that allow you to send selected findings and insight results to EventBridge.

You then configure EventBridge rules to respond to each type of event.

For more information about using EventBridge, see the [Amazon EventBridge User Guide](#).

Note

As a best practice, make sure that the permissions granted to your users to access EventBridge use least-privilege IAM policies that grant only the required permissions.

For more information, see [Identity and access management in Amazon EventBridge](#).

A set of templates for cross-account automated response and remediation is also available in AWS Solutions. The templates leverage EventBridge event rules and Lambda functions. You deploy the solution using AWS CloudFormation and AWS Systems Manager. The solution can create fully automated response and remediation actions. It can also use Security Hub custom actions to create user-triggered response and remediation actions. For details on how to configure and use the solution, see the [AWS Solutions page for the Security Hub Automated Response and Remediation Solution](#).

Topics

- [Types of Security Hub integration with EventBridge \(p. 604\)](#)
- [EventBridge event formats for Security Hub \(p. 606\)](#)
- [Configuring an EventBridge rule for automatically sent findings \(p. 608\)](#)
- [Using custom actions to send findings and insight results to EventBridge \(p. 612\)](#)

Types of Security Hub integration with EventBridge

Security Hub uses the following EventBridge event types to support the following types of integration with EventBridge.

On the EventBridge dashboard for Security Hub, **All Events** includes all of these event types.

All findings (Security Hub Findings - Imported)

Security Hub automatically sends all new findings and all updates to existing findings to EventBridge as **Security Hub Findings - Imported** events. Each **Security Hub Findings - Imported** event contains a single finding.

Every [BatchImportFindings](#) and [BatchUpdateFindings](#) request triggers a **Security Hub Findings - Imported** event.

For administrator accounts, the event feed in EventBridge includes events for findings from both their account and from their member accounts.

In an aggregation Region, the event feed includes events for findings from the aggregation Region and the linked Regions. Cross-Region findings are included in the event feed in near real time. For information on how to configure finding aggregation, see [Cross-Region aggregation \(p. 111\)](#).

You can define rules in EventBridge that automatically route findings to an Amazon S3 bucket, a remediation workflow, or a third-party tool. The rules can include filters that only apply the rule if the finding has specific attribute values.

You use this method to automatically send all findings, or all findings that have specific characteristics, to a response or remediation workflow.

See [the section called "Configuring a rule for automatically sent findings" \(p. 608\)](#).

Findings for custom actions (Security Hub Findings - Custom Action)

Security Hub also sends findings that are associated with custom actions to EventBridge as **Security Hub Findings - Custom Action** events.

This is useful for analysts working with the Security Hub console who want to send a specific finding, or a small set of findings, to a response or remediation workflow. You can select a custom action for up to 20 findings at a time. Each finding is sent to EventBridge as a separate EventBridge event.

When you create a custom action, you assign it a custom action ID. You can use this ID to create an EventBridge rule that takes a specified action after receiving a finding that is associated with that custom action ID.

See [the section called "Configuring and using custom actions" \(p. 612\)](#).

For example, you can create a custom action in Security Hub called `send_to_ticketing`. Then in EventBridge, you create a rule that is triggered when EventBridge receives a finding that includes the `send_to_ticketing` custom action ID. The rule includes logic to send the finding to your ticketing system. You can then select findings within Security Hub and use the custom action in Security Hub to manually send findings to your ticketing system.

For examples of how to send Security Hub findings to EventBridge for further processing, see [How to Integrate AWS Security Hub Custom Actions with PagerDuty](#) and [How to Enable Custom Actions in AWS Security Hub](#) on the AWS Partner Network (APN) Blog.

Insight results for custom actions (Security Hub Insight Results)

You can also use custom actions to send sets of insight results to EventBridge as **Security Hub Insight Results** events. Insight results are the resources that match an insight. Note that when you send

insight results to EventBridge, you are not sending the findings to EventBridge. You are only sending the resource identifiers that are associated with the insight results. You can send up to 100 resource identifiers at a time.

Similar to custom actions for findings, you first create the custom action in Security Hub, and then create a rule in EventBridge.

See [the section called "Configuring and using custom actions" \(p. 612\)](#).

For example, suppose you see a particular insight result of interest that you want to share with a colleague. In that case, you can use a custom action to send that insight result to the colleague through a chat or ticketing system.

EventBridge event formats for Security Hub

The **Security Hub Findings - Imported**, **Security Findings - Custom Action**, and **Security Hub Insight Results** event types use the following event formats.

The event format is the format that is used when Security Hub sends an event to EventBridge.

Security Hub Findings - Imported

Security Hub Findings - Imported events that are sent from Security Hub to EventBridge use the following format.

```
{
  "version": "0",
  "id": "CWE-event-id",
  "detail-type": "Security Hub Findings - Imported",
  "source": "aws.securityhub",
  "account": "111122223333",
  "time": "2019-04-11T21:52:17Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:securityhub:us-west-2::product/aws/maciek/arn:aws:maciek:us-
west-2:111122223333:integtest/trigger/6294d71b927c41cbab915159a8f326a3/alert/f2893b211841"
  ],
  "detail": {
    "findings": [
      {
        <finding content>
      }
    ]
  }
}
```

<finding content> is the content, in JSON format, of the finding that is sent by the event. Each event sends a single finding.

For a complete list of finding attributes, see [AWS Security Finding Format \(ASFF\) \(p. 132\)](#).

For information about how to configure EventBridge rules that are triggered by these events, see [the section called "Configuring a rule for automatically sent findings" \(p. 608\)](#).

Security Hub Findings - Custom Action

Security Hub Findings - Custom Action events that are sent from Security Hub to EventBridge use the following format. Each finding is sent in a separate event.

```
{
```

```
"version": "0",
"id": "1a1111a1-b22b-3c33-444d-5555e5ee5555",
"detail-type": "Security Hub Findings - Custom Action",
"source": "aws.securityhub",
"account": "111122223333",
"time": "2019-04-11T18:43:48Z",
"region": "us-west-1",
"resources": [
  "arn:aws:securityhub:us-west-1:111122223333:action/custom/custom-action-name"
],
"detail": {
  "actionName": "custom-action-name",
  "actionDescription": "description of the action",
  "findings": [
    {
      <finding content>
    }
  ]
}
```

`<finding content>` is the content, in JSON format, of the finding that is sent by the event. Each event sends a single finding.

For a complete list of finding attributes, see [AWS Security Finding Format \(ASFF\)](#) (p. 132).

For information about how to configure EventBridge rules that are triggered by these events, see [the section called "Configuring and using custom actions"](#) (p. 612).

Security Hub Insight Results

Security Hub Insight Results events that are sent from Security Hub to EventBridge use the following format.

```
{
  "version": "0",
  "id": "1a1111a1-b22b-3c33-444d-5555e5ee5555",
  "detail-type": "Security Hub Insight Results",
  "source": "aws.securityhub",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:securityhub:us-west-1:111122223333::product/aws/macie:us-west-1:222233334444:test/trigger/1ec9cf700ef6be062b19584e0b7d84ec/alert/f2893b211841"
  ],
  "detail": {
    "actionName": "name of the action",
    "actionDescription": "description of the action",
    "insightArn": "ARN of the insight",
    "insightName": "Name of the insight",
    "resultType": "ResourceAwsIamAccessKeyUserName",
    "number of results": "number of results, max of 100",
    "insightResults": [
      {"result 1": 5},
      {"result 2": 6}
    ]
  }
}
```

For information about how to create an EventBridge rule that is triggered by these events, see [the section called "Configuring and using custom actions"](#) (p. 612).

Configuring an EventBridge rule for automatically sent findings

You can create a rule in EventBridge that defines an action to take when a **Security Hub Findings - Imported** event is received. **Security Hub Findings - Imported** events are triggered by updates from both [BatchImportFindings](#) and [BatchUpdateFindings](#).

Each rule contains an event pattern, which identifies the events that trigger the rule. The event pattern always contains the event source (`aws.securityhub`) and the event type (**Security Hub Findings - Imported**). The event pattern can also specify filters to identify the findings that the rule applies to.

The rule then identifies the rule targets. The targets are the actions to take when EventBridge receives a **Security Hub Findings - Imported** event and the finding matches the filters.

The instructions provided here use the EventBridge console. When you use the console, EventBridge automatically creates the required resource-based policy that enables EventBridge to write to CloudWatch Logs.

You can also use the `PutRule` API operation of the EventBridge API. However, if you use the EventBridge API, then you must create the resource-based policy. For details on the required policy, see [CloudWatch Logs permissions](#) in the *Amazon EventBridge User Guide*.

Format of the event pattern

The format of the event pattern for **Security Hub Findings - Imported** events is as follows:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Findings - Imported"
  ],
  "detail": {
    "findings": {
      <attribute filter values>
    }
  }
}
```

- `source` identifies Security Hub as the service that generates the event.
- `detail-type` identifies the type of event.
- `detail` is optional and provides the filter values for the event pattern. If the event pattern does not contain a `detail` field, then all findings trigger the rule.

You can filter the findings based on any finding attribute. For each attribute, you provide a comma-separated array of one or more values.

```
"<attribute name>": [ "<value1>", "<value2>" ]
```

If you provide more than one value for an attribute, then those values are joined by `OR`. A finding matches the filter for an individual attribute if the finding has any of the listed values. For example, if you provide both `INFORMATIONAL` and `LOW` as values for `Severity.Label`, then the finding matches if it has a severity label of either `INFORMATIONAL` or `LOW`.

The attributes are joined by **AND**. A finding matches if it matches the filter criteria for all of the provided attributes.

When you provide an attribute value, it must reflect the location of that attribute within the AWS Security Finding Format (ASFF) structure.

In the following example, the event pattern provides filter values for `ProductArn` and `Severity.Label`, so a finding matches if it is generated by Amazon Inspector and it has a severity label of either `INFORMATIONAL` or `LOW`.

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Findings - Imported"
  ],
  "detail": {
    "findings": {
      "ProductArn": ["arn:aws:securityhub:us-east-1:product/aws/inspector"],
      "Severity": {
        "Label": ["INFORMATIONAL", "LOW"]
      }
    }
  }
}
```

Creating an event rule

You can use a predefined event pattern or a custom event pattern to create a rule in EventBridge. If you select a predefined pattern, EventBridge automatically fills in `source` and `detail-type`. EventBridge also provides fields to specify filter values for the following finding attributes:

- `AwsAccountId`
- `Compliance.Status`
- `Criticality`
- `ProductArn`
- `RecordState`
- `ResourceId`
- `ResourceType`
- `Severity.Label`
- `Types`
- `Workflow.Status`

To create an EventBridge rule

1. Open the Amazon EventBridge console at <https://console.aws.amazon.com/events/>.
2. Using the following values, create an EventBridge rule that monitors finding events:
 - For **Rule type**, choose **Rule with an event pattern**.
 - Choose how to build the event pattern.

To build the event pattern with...	Do this...	
A template	<p>In the Event pattern section, choose the following options:</p> <ul style="list-style-type: none">• For Event source, choose AWS services.• For AWS service, choose Security Hub.• For Event type, choose Security Hub Findings - Imported.• (Optional) To make the rule more specific, add filter values. For example, to limit the rule to findings with active record states, for Specific Record state(s), choose Active.	

To build the event pattern with...	Do this...	
<p>A custom event pattern</p> <p>(Use a custom pattern if you want to filter findings based on attributes that do not appear in the EventBridge console.)</p>	<ul style="list-style-type: none"> In the Event pattern section, choose Custom patterns (JSON editor), and then paste the following event pattern into the text area: <pre data-bbox="781 499 1114 940"> { "source": ["aws.securityhub"], "detail-type": ["Security Hub Findings - Imported"], "detail": { "findings": { "<attribute name>": ["<value1>", "<value2>"] } } } </pre> Update the event pattern to include the attribute and attribute values that you want to use as a filter. <p>For example, to apply the rule to findings that have a verification state of <code>TRUE_POSITIVE</code>, use the following pattern example:</p> <pre data-bbox="781 1262 1114 1696"> { "source": ["aws.securityhub"], "detail-type": ["Security Hub Findings - Imported"], "detail": { "findings": { "VerificationState": ["TRUE_POSITIVE"] } } } </pre> 	

- For **Target types**, choose **AWS service**, and for **Select a target**, choose a target such as an Amazon SNS topic or AWS Lambda function. The target is triggered when an event is received that matches the event pattern defined in the rule.

For details about creating rules, see [Creating Amazon EventBridge rules that react to events](#) in the *Amazon EventBridge User Guide*.

Using custom actions to send findings and insight results to EventBridge

To use Security Hub custom actions to send findings or insight results to EventBridge, you first create the custom action in Security Hub. Then define the rule in EventBridge.

You can create up to 50 custom actions.

If you enabled cross-Region aggregation, and manage findings from the aggregation Region, then create custom actions in the aggregation Region.

The rule in EventBridge uses the ARN from the custom action.

Creating a custom action (console)

When you create a custom action, you specify the name, description, and a unique identifier.

To create a custom action in Security Hub (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings** and then choose **Custom actions**.
3. Choose **Create custom action**.
4. Provide a **Name**, **Description**, and **Custom action ID** for the action.

The **Name** must be fewer than 20 characters.

The **Custom action ID** must be unique for each AWS account.

5. Choose **Create custom action**.
6. Make a note of the **Custom action ARN**. You need to use the ARN when you create a rule to associate with this action in EventBridge.

Creating a custom action (Security Hub API, AWS CLI)

To create a custom action, you can use an API call or the AWS Command Line Interface.

To create a custom action (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `CreateActionTarget` operation. When you create a custom action, you provide the name, description, and custom action identifier.
- **AWS CLI** – At the command line, run the `create-action-target` command.

```
create-action-target --name <customActionName> --description <customActionDescription> --id <customActionIdentifier>
```

Example

```
aws securityhub create-action-target --name "Send to remediation" --description "Action to send the finding for remediation tracking" --id "Remediation"
```

Defining a rule in EventBridge

To process the custom action, you must create a corresponding rule in EventBridge. The rule definition includes the ARN of the custom action.

The event pattern for a **Security Hub Findings - Custom Action** event has the following format:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Findings - Custom Action"
  ],
  "resources": [ "<custom action ARN>" ]
}
```

The event pattern for a **Security Hub Insight Results** event has the following format:

```
{
  "source": [
    "aws.securityhub"
  ],
  "detail-type": [
    "Security Hub Insight Results"
  ],
  "resources": [ "<custom action ARN>" ]
}
```

In both patterns, *<custom action ARN>* is the ARN of a custom action. You can configure a rule that applies to more than one custom action.

The instructions provided here are for the EventBridge console. When you use the console, EventBridge automatically creates the required resource-based policy that enables EventBridge to write to CloudWatch Logs.

You can also use the [PutRule](#) API operation of the EventBridge API. However, if you use the EventBridge API, then you must create the resource-based policy. For details on the required policy, see [CloudWatch Logs permissions](#) in the *Amazon EventBridge User Guide*.

To define a rule in EventBridge

1. Open the Amazon EventBridge console at <https://console.aws.amazon.com/events/>.
2. In the navigation pane, choose **Rules**.
3. Choose **Create rule**.
4. Enter a name and description for the rule.
5. For **Event bus**, choose the event bus that you want to associate with this rule. If you want this rule to match events that come from your account, select **default**. When an AWS service in your account emits an event, it always goes to your account's default event bus.
6. For **Rule type**, choose **Rule with an event pattern**.
7. Choose **Next**.

8. For **Event source**, choose **AWS events**.
9. For **Event pattern**, choose **Event pattern form**.
10. For **Event source**, choose **AWS services**.
11. For **AWS service**, choose **Security Hub**.
12. For **Event type**, do one of the following:
 - To create a rule to apply when you send findings to a custom action, choose **Security Hub Findings - Custom Action**.
 - To create a rule to apply when you send insight results to a custom action, choose **Security Hub Insight Results**.
13. Choose **Specific custom action ARNs**, add a custom action ARN.

If the rule applies to multiple custom actions, choose **Add** to add more custom action ARNs.
14. Choose **Next**.
15. Under **Select targets**, choose and configure the target to invoke when this rule is matched.
16. Choose **Next**.
17. (Optional) Enter one or more tags for the rule. For more information, see [Amazon EventBridge tags](#) in the *Amazon EventBridge User Guide*.
18. Choose **Next**.
19. Review the details of the rule and choose **Create rule**.

When you perform a custom action on findings or insight results in your account, events are generated in EventBridge.

Selecting a custom action for findings and insight results

After you create your Security Hub custom actions and EventBridge rules, you can send findings and insight results to EventBridge for further management and processing.

Events are sent to EventBridge only in the account in which they are viewed. If you view a finding using an administrator account, the event is sent to EventBridge in the administrator account.

For AWS API calls to be effective, the implementations of target code must switch roles into member accounts. This also means that the role you switch into must be deployed to each member where action is needed.

To send findings to EventBridge

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. Display a list of findings:
 - From **Findings**, you can view findings from all of the enabled product integrations and controls.
 - From **Security standards**, you can navigate to a list of findings generated from a selected control. See [the section called "Viewing details for a control"](#) (p. 339).
 - From **Integrations**, you can navigate to a list of findings generated by an enabled integration. See [the section called "Viewing the findings from an integration"](#) (p. 293).
 - From **Insights**, you can navigate to a list of findings for an insight result. See [the section called "Viewing insight results and findings"](#) (p. 277).
3. Select the findings to send to EventBridge. You can select up to 20 findings at a time.
4. From **Actions**, choose the custom action that aligns with the EventBridge rule to apply.

Security Hub sends a separate **Security Hub Findings - Custom Action** event for each finding.

To send insight results to EventBridge

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Insights**.
3. On the **Insights** page, choose the insight that includes the results to send to EventBridge.
4. Select the insight results to send to EventBridge. You can select up to 20 results at a time.
5. From **Actions**, choose the custom action that aligns with the EventBridge rule to apply.

Disabling AWS Security Hub

To disable AWS Security Hub, you can use the Security Hub console or the Security Hub API.

You cannot disable Security Hub in the following cases:

- Your account is the designated Security Hub administrator account for an organization.
- Your account is a Security Hub administrator account by invitation, and you have member accounts that are enabled. Before you can disable Security Hub, you must disassociate all of your member accounts. See [the section called “Disassociating member accounts” \(p. 106\)](#).
- Your account is a member account. Before you can disable Security Hub, your account must be disassociated from your administrator account.

For an organization account, only the administrator account can disassociate member accounts. See [the section called “Disassociating member accounts” \(p. 101\)](#).

For manually invited accounts, either the administrator account or the member account can disassociate the member account. See [the section called “Disassociating member accounts” \(p. 106\)](#) or [the section called “Disassociating from your administrator account” \(p. 107\)](#).

When you disable Security Hub for an account, it is disabled only in the current Region. No new findings are processed for the account in that Region.

The following also occurs.

- After 90 days, your existing findings and insights and any Security Hub configuration settings are deleted and cannot be recovered.

If you want to save your existing findings, you must export them before you disable Security Hub. For more information, see [the section called “Effect of account actions on Security Hub data” \(p. 108\)](#).

- Any enabled standards are disabled.

Disabling Security Hub (console)

You can disable Security Hub from the AWS Management Console.

To disable Security Hub (console)

1. Open the AWS Security Hub console at <https://console.aws.amazon.com/securityhub/>.
2. In the navigation pane, choose **Settings**.
3. On the **Settings** page, choose **General**.
4. Under **Disable AWS Security Hub**, choose **Disable AWS Security Hub**. Then choose **Disable AWS Security Hub** again.

Disabling Security Hub (Security Hub API, AWS CLI)

To disable Security Hub, you can use an API call or the AWS Command Line Interface.

To disable Security Hub (Security Hub API, AWS CLI)

- **Security Hub API** – Use the `DisableSecurityHub` operation.
- **AWS CLI** – At the command line, run the `disable-security-hub` command.

```
aws securityhub disable-security-hub
```

Document history for the AWS Security Hub User Guide

The following table describes the updates to the documentation for AWS Security Hub.

update-history-change	update-history-description	update-history-date
AWS Security Hub adds new security controls	<p>The following 36 new Security Hub controls are available to customers who have enabled the FSBP standard. Some controls have Regional limitations.</p> <ul style="list-style-type: none"> the section called “[AutoScaling.3] Auto Scaling group should configure EC2 instances to require Instance Metadata Service Version 2 (IMDSv2)” (p. 465) the section called “[AutoScaling.4] Auto Scaling group launch configuration should not have metadata response hop limit greater than 1” (p. 466) the section called “[AutoScaling.6] Auto Scaling groups should use multiple instance types in multiple Availability Zones” (p. 468) the section called “[CloudFormation.1] CloudFormation stacks should be integrated with Simple Notification Service (SNS)” (p. 469) the section called “[CloudFront.10] CloudFront distributions should not use deprecated SSL protocols between edge locations and custom origins” (p. 474) the section called “[EC2.23] EC2 Transit Gateways should not automatically accept VPC attachment requests” (p. 499) the section called “[EC2.24] Paravirtual EC2 instance types should not be used” (p. 500) 	June 22, 2022

- the section called “[EC2.27] Running EC2 Instances should not use key pairs” (p. 501)
- the section called “[ECR.1] ECR private repositories should have image scanning configured” (p. 502)
- the section called “[ECR.2] ECR private repositories should have tag immutability configured” (p. 502)
- the section called “[ECS.3] ECS task definitions should not share the host's process namespace” (p. 505)
- the section called “[ECS.4] ECS containers should run as non-privileged” (p. 505)
- the section called “[ECS.5] ECS containers should be limited to read-only access to root filesystems” (p. 506)
- the section called “[ECS.8] Secrets should not be passed as container environment variables” (p. 507)
- the section called “[ECS.10] Fargate services should run on the latest Fargate platform version” (p. 508)
- the section called “[ECS.12] ECS clusters should have Container Insights enabled” (p. 508)
- the section called “[EFS.3] EFS access points should enforce a root directory” (p. 510)
- the section called “[EFS.4] EFS access points should enforce a user identity” (p. 511)
- the section called “[EKS.2] EKS clusters should run on a supported Kubernetes version” (p. 512)
- the section called “[ELB.12] Application Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 520)
- the section called “[ELB.13] Application, Network, and Gateway Load Balancers should span multiple Availability Zones” (p. 520)

- the section called “[ELB.14] Classic Load Balancers should be configured with defensive or strictest desync mitigation mode” (p. 521)
- the section called “[Kinesis.1] Kinesis Data Streams should be encrypted at rest” (p. 537)
- the section called “[Network Firewall.3] Network Firewall policies should have at least one rule group associated” (p. 543)
- the section called “[NetworkFirewall.4] The default stateless action for Network Firewall policies should be drop or forward for full packets” (p. 543)
- the section called “[NetworkFirewall.5] The default stateless action for Network Firewall policies should be drop or forward for fragmented packets” (p. 544)
- the section called “[OpenSearch.7] OpenSearch domains should have fine-grained access control enabled” (p. 548)
- the section called “[Redshift.9] Redshift clusters should not use the default database name” (p. 574)
- the section called “[S3.13] S3 buckets should have lifecycle policies configured” (p. 583)
- the section called “[SNS.2] Logging of delivery status should be enabled for notification messages sent to a topic” (p. 589)
- the section called “[WAF.2] A WAF Regional rule should have at least one condition” (p. 595)
- the section called “[WAF.3] A WAF Regional rule group should have at least one rule” (p. 596)
- the section called “[WAF.4] A WAF Classic Regional web ACL should have at least one rule or rule group” (p. 596)

	<ul style="list-style-type: none"> the section called “[WAF.6] A WAF global rule should have at least one condition” (p. 597) the section called “[WAF.7] A WAF global rule group should have at least one rule” (p. 597) the section called “[WAF.8] A WAF global web ACL should have at least one rule or rule group” (p. 598) 	
AWS Security Hub supports a new Region	Security Hub is now available in Asia Pacific (Jakarta). Some controls are not available in this Region.	June 7, 2022
Improved integration between AWS Security Hub and AWS Config	Security Hub users can see the results of AWS Config rule evaluations as findings in Security Hub.	June 6, 2022
Added ability to opt out of auto-enabled standards	For users who have integrated with AWS Organizations, this feature allows you to log into the Security Hub administrator account and opt new member accounts out of auto-enabled standards.	April 25, 2022
Expanded cross-Region aggregation	Added cross-Region aggregation to control statuses and security scores.	April 20, 2022
CompanyName and ProductName are now top level attributes	Added new top level attributes for setting company and product names associated with custom integrations	April 1, 2022
Added new controls to the AWS Foundational Security Best Practices standard	Added 5 new controls to the AWS Foundational Security Best Practices standard.	March 31, 2022
Added new resource details objects to ASFF	Added <code>AwsRdsDbSecurityGroup</code> resource type to ASFF.	March 25, 2022
Added additional resources details in ASFF	Added additional details to <code>AwsAutoScalingScalingGroup</code> , <code>AwsElbLoadBalancer</code> , <code>AwsRedshiftCluster</code> , and <code>AwsCodeBuildProject</code> .	March 25, 2022
Added new controls to the AWS Foundational Security Best Practices standard	Added 15 new controls to the AWS Foundational Security Best Practices standard.	March 16, 2022

Added new controls to the AWS Foundational Security Best Practices standard and Payment Card Industry Data Security Standard (PCI DSS)	Added new controls for Amazon OpenSearch Service, Amazon RDS, Amazon EC2, Elastic Load Balancing, and CloudFront to the AWS Foundational Security Best Practices standard. Also added two new controls for OpenSearch Service to the PCI DSS.	February 15, 2022
Added new field to ASFF	Added new field: Sample.	January 26, 2022
Added integration with AWS Health	AWS Health uses service-to-service event messaging to send findings to Security Hub.	January 19, 2022
Added integration with AWS Trusted Advisor	Trusted Advisor sends the results of its checks to Security Hub as Security Hub findings. Security Hub sends the results of its AWS Foundational Security Best Practices checks to Trusted Advisor.	January 18, 2022
Updated resource details objects in ASFF	Added <code>MixedInstancesPolicy</code> and <code>AvailabilityZones</code> to <code>AwsAutoScalingAutoScalingGroup</code> . Added <code>MetadataOptions</code> to <code>AwsAutoScalingLaunchConfiguration</code> . Added <code>BucketVersioningConfiguration</code> to <code>AwsS3Bucket</code> .	December 20, 2021
Updated output for ASFF documentation	The descriptions of ASFF attributes were previously in a single topic. Each top-level object and each resource details object is now in its own topic. The ASFF syntax topic contains links to those topics.	December 20, 2021
Added new resource details objects to ASFF for AWS Network Firewall	For AWS Network Firewall, added the following resource details objects: <code>AwsNetworkFirewallFirewall</code> , <code>AwsNetworkFireFirewallPolicy</code> , and <code>AwsNetworkFirewallRuleGroup</code> .	December 20, 2021
Added support for the new version of Amazon Inspector	Security Hub is integrated with the new version of Amazon Inspector as well as with Amazon Inspector Classic. Amazon Inspector sends findings to Security Hub.	November 29, 2021

Changed the severity of EC2.19	The severity of EC2.19 (Security groups should not allow unrestricted access to ports with high risk) is changed from High to Critical.	November 17, 2021
New integration with Sonrai Dig	Security Hub now offers an integration with Sonrai Dig. Sonrai Dig monitors cloud environments to identify security risks. Sonrai Dig sends findings to Security Hub.	November 12, 2021
Updated check for CIS 2.1 and CloudTrail.1 controls (p. 618)	In addition to checking that at least one multi-Region CloudTrail trail is in place, CIS 2.1 and CloudTrail.1 now also check that the <code>ExcludeManagementEventSources</code> parameter is empty in at least one of the multi-Region CloudTrail trails.	November 9, 2021
Added support for VPC endpoints	Security Hub is now integrated with AWS PrivateLink and supports VPC endpoints.	November 3, 2021
Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Elastic Load Balancing (ELB.2 and ELB.8) and AWS Systems Manager (SSM.4).	November 2, 2021
Added ports to the check for the EC2.19 control	EC2.19 now also checks that security groups do not allow unrestricted ingress access to the following ports: 3000 (Go, Node.js, and Ruby web development frameworks), 5000 (Python web development frameworks), 8088 (legacy HTTP port), and 8888 (alternative HTTP port)	October 27, 2021
Added the integration with Logz.io Cloud SIEM	Logz.io is a provider of Cloud SIEM that provides advanced correlation of log and event data to help security teams to detect, analyze, and respond to security threats in real time. Logz.io receives findings from Security Hub.	October 25, 2021

Added support for cross-Region aggregation of findings	Cross-Region aggregation allows you to view all of your findings without having to change Regions. Administrator accounts choose an aggregation Region and linked Regions. Findings for the administrator account and its member accounts are aggregated from the linked Regions to the aggregation Region.	October 20, 2021
Updated resource details objects in ASFF	Added viewer certificate details to <code>AwsCloudFrontDistribution</code> . Added additional details to <code>AwsCodeBuildProject</code> . Added load balancer attributes to <code>AwsElbV2LoadBalancer</code> . Added the S3 bucket owner account identifier to <code>AwsS3Bucket</code> .	October 8, 2021
Added new resource details objects to ASFF	Added the following new resource details objects to ASFF: <code>AwsEc2VpcEndpointService</code> , <code>AwsEcrRepository</code> , <code>AwsEksCluster</code> , <code>AwsOpenSearchServiceDomain</code> , <code>AwsWafRateBasedRule</code> , <code>AwsWafRegionalRateBasedRule</code> , <code>AwsXrayEncryptionConfig</code>	October 8, 2021
Removed deprecated runtime from the Lambda.2 control	In the AWS Foundational Security Best Practices standard, removed the <code>dotnetcore2.1</code> runtime from [Lambda.2] Lambda functions should use supported runtimes .	October 6, 2021
New name for Check Point integration	The integration with Check Point Dome9 Arc is now Check Point CloudGuard Posture Management. The integration ARN did not change.	October 1, 2021
Removed the integration with Alcide (p. 618)	The integration with Alcide <code>kAudit</code> is discontinued.	September 30, 2021
Changed the severity of EC2.19	The severity of [EC2.19] Security groups should not allow unrestricted access to ports with high risk is changed from Medium to High.	September 30, 2021

Integration with AWS Organizations is now supported in the China Regions (p. 618)	The Security Hub integration with Organizations is now supported in China (Beijing) and China (Ningxia).	September 20, 2021
New AWS Config rule for the S3.1 and PCI.S3.6 controls (p. 618)	Both S3.1 and PCI.S3.6 verify that the Amazon S3 Block Public Access setting is enabled. The AWS Config rule for these controls is changed from <code>s3-account-level-public-access-blocks</code> to <code>s3-account-level-public-access-blocks-periodic</code> .	September 14, 2021
Removed deprecated runtimes from the Lambda.2 control	In the AWS Foundational Security Best Practices standard, removed the <code>nodejs10.x</code> and <code>ruby2.5</code> runtimes from [Lambda.2] Lambda functions should use supported runtimes.	September 13, 2021
Changed the severity of the CIS 2.2 control	In the CIS AWS Foundations Benchmark standard, the severity for 2.2. – Ensure CloudTrail log file validation is enabled is changed from Low to Medium.	September 13, 2021
Updated ECS.1, Lambda.2, and SSM.1 in the AWS Foundational Security Best Practices standard	In the AWS Foundational Security Best Practices standard, ECS.1 now has a <code>SkipInactiveTaskDefinitions</code> parameter that is set to <code>true</code> . This ensures that the control only checks active task definitions. For Lambda.2, added Python 3.9 to the list of runtimes. SSM.1 now checks both stopped and running instances.	September 7, 2021
PCI.Lambda.2 control now excludes Lambda@Edge resources	In the Payment Card Industry Data Security Standard (PCI DSS) standard, the PCI.Lambda.2 control now excludes Lambda@Edge resources.	September 7, 2021
Added the integration with HackerOne Vulnerability Intelligence	Security Hub now offers an integration with HackerOne Vulnerability Intelligence. The integration sends findings to Security Hub.	September 7, 2021

Updated resource details objects in ASFF	For <code>AwsKmsKey</code> , added <code>KeyRotationStatus</code> . For <code>AwsS3Bucket</code> , added <code>AccessControlList</code> , <code>BucketLoggingConfiguration</code> , <code>BucketNotificationConfiguration</code> , and <code>BucketWebsiteConfiguration</code> .	September 2, 2021
Added new resource details objects to ASFF	Added the following new resource details objects to ASFF: <code>AwsAutoScalingLaunchConfiguration</code> , <code>AwsEc2VpnConnection</code> , and <code>AwsEcrContainerImage</code> .	September 2, 2021
Added details to the Vulnerabilities object in ASFF	In <code>Cvss</code> , added <code>Adjustments</code> and <code>Source</code> . In <code>VulnerablePackages</code> , added the file path and package manager.	September 2, 2021
Systems Manager Explorer and OpsCenter integration now supported in the China Regions (p. 618)	The Security Hub integration with SSM Explorer and OpsCenter is now supported in China (Beijing) and China (Ningxia).	August 31, 2021
Retiring the Lambda.4 control (p. 618)	Security Hub is retiring the control [Lambda.4] Lambda functions should have a dead-letter queue configured . When a control is retired, it no longer displays on the console, and Security Hub does not perform checks against it.	August 31, 2021
Retiring the PCI.EC2.3 control (p. 618)	Security Hub is retiring the control [PCI.EC2.3] Unused EC2 security groups should be removed . When a control is retired, it no longer displays on the console, and Security Hub does not perform checks against it.	August 27, 2021
Change to how Security Hub sends findings to custom actions	When you send findings to a custom action, Security Hub now sends each finding in a separate Security Hub Findings - Custom Action event.	August 20, 2021
Added a new compliance status reason code for custom Lambda runtimes	Added a new <code>LAMBDA_CUSTOM_RUNTIME_DETAILS_NOT_AVAILABLE</code> compliance status reason code. This reason code indicates that Security Hub could not perform a check against a custom Lambda runtime.	August 20, 2021

AWS Firewall Manager integration now supported in the China Regions (p. 618)	The Security Hub integration with Firewall Manager is now supported in China (Beijing) and China (Ningxia).	August 19, 2021
New integrations with Caveonix Cloud and Forcepoint Cloud Security Gateway	Security Hub now offers integrations with Caveonix Cloud and Forcepoint Cloud Security Gateway. Both integrations send findings to Security Hub.	August 10, 2021
Added new <code>CompanyName</code> , <code>ProductName</code> , and <code>Region</code> attributes to ASFF	Added <code>CompanyName</code> , <code>ProductName</code> , and <code>Region</code> fields to the top level of the ASFF. These fields are populated automatically and, except for custom product integrations, cannot be updated using <code>BatchImportFindings</code> or <code>BatchUpdateFindings</code> . On the console, finding filters use these new fields. In the API, the <code>CompanyName</code> and <code>ProductName</code> filters use the attributes that are under <code>ProductFields</code> .	July 23, 2021
Added and updated resource details objects in ASFF	Added a new <code>AwsRdsEventSubscription</code> resource type and resource details. Added resource details for the <code>AwsEcsService</code> resource type. Added attributes to the <code>AwsElasticsearchDomain</code> resource details object.	July 23, 2021
Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.5), Amazon EC2 (EC2.19), Amazon ECS (ECS.2), Elastic Load Balancing (ELB.7), Amazon OpenSearch Service (ES.5 through ES.8), Amazon RDS (RDS.16 through RDS.23), Amazon Redshift (Redshift.4), and Amazon SQS (SQS.1).	July 20, 2021
Moved a permission within the service-linked role managed policy	Moved the <code>config:PutEvaluations</code> permission within the managed policy <code>AWSecurityHubServiceRolePolicy</code> , so that it is applied to all resources.	July 14, 2021

Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.4), Amazon CloudFront (CloudFront.5 and CloudFront.6), Amazon EC2 (EC2.17 and EC2.18), Amazon ECS (ECS.1), Amazon OpenSearch Service (ES.4), AWS Identity and Access Management (IAM.21), Amazon RDS (RDS.15), and Amazon S3 (S3.8).	July 8, 2021
Added new compliance status reason codes for control findings	INTERNAL_SERVICE_ERROR indicates that an unknown error occurred. SNS_TOPIC_CROSS_ACCOUNT indicates that the SNS topic is owned by a different account. SNS_TOPIC_INVALID indicates that the associated SNS topic is invalid.	July 6, 2021
Added the integration with AWS Chatbot	Added the integration with AWS Chatbot. Security Hub sends findings to AWS Chatbot.	June 30, 2021
Added a new permission to the service-linked role managed policy	Added a new permission to the managed policy <code>AWSecurityHubServiceRolePolicy</code> to allow the service-linked role to deliver evaluation results to AWS Config.	June 29, 2021
New and updated resource details objects in the ASFF	Added new resource details objects for ECS clusters and ECS task definitions. Updated the EC2 instance object to list the associated network interfaces. Added the client certificate ID for the API Gateway V2 stages. Added the lifecycle configuration for S3 buckets.	June 24, 2021
Updated the calculation of aggregated control statuses and standard security scores (p. 618)	Security Hub now calculates the overall control status and standard security score every 24 hours. For administrator accounts, the score now reflects whether each control is enabled or disabled for each account.	June 23, 2021
Updated information about Security Hub handling of suspended accounts	Added information on how Security Hub handles accounts that are suspended in AWS.	June 23, 2021

<p>Added tabs to display the enabled and disabled controls for the individual administrator account</p>	<p>For the administrator account, the main tabs on the standard details page contain aggregated information across accounts. The new Enabled for this account and Disabled for this account tabs list the accounts that are enabled or disabled for the individual administrator account.</p>	<p>June 23, 2021</p>
<p>Added <code>java8.a12</code> to the parameters for <code>Lambda.2</code></p>	<p>In the AWS Foundational Security Best Practices standard, added <code>java8.a12</code> to the supported runtimes for the <code>Lambda.2</code> control.</p>	<p>June 8, 2021</p>
<p>New integrations with MicroFocus ArcSight and NETSCOUT Cyber Investigator</p>	<p>Added integrations with MicroFocus ArcSight and NETSCOUT Cyber Investigator. MicroFocus ArcSight receives findings from Security Hub. NETSCOUT Cyber Investigator sends findings to Security Hub.</p>	<p>June 7, 2021</p>
<p>Added details for <code>AWSecurityHubServiceRolePolicy</code></p>	<p>Updated the managed policies section to add details for the existing managed policy <code>AWSecurityHubServiceRolePolicy</code>, which is used by the Security Hub service-linked role.</p>	<p>June 4, 2021</p>
<p>New integration with Jira Service Management</p>	<p>The AWS Service Management Connector for Jira sends findings to Jira and uses them to create Jira issues. When the Jira issues are updated, the corresponding findings in Security Hub also are updated.</p>	<p>May 26, 2021</p>
<p>Updated the supported controls list for the Asia Pacific (Osaka) Region</p>	<p>Updated the CIS AWS Foundations standard and the Payment Card Industry Data Security Standard (PCI DSS) to indicate the controls that are not supported in Asia Pacific (Osaka).</p>	<p>May 21, 2021</p>
<p>New integration with Sysdig Secure for cloud</p>	<p>Added an integration with Sysdig Secure for cloud. The integration sends findings to Security Hub.</p>	<p>May 14, 2021</p>

Added controls to the AWS Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.2 and APIGateway.3), AWS CloudTrail (CloudTrail.4 and CloudTrail.5), Amazon EC2 (EC2.15 and EC2.16), AWS Elastic Beanstalk (ElasticBeanstalk.1 and ElasticBeanstalk.2), AWS Lambda (Lambda.4), Amazon RDS (RDS.12 – RDS.14), Amazon Redshift (Redshift.7), AWS Secrets Manager (SecretsManager.3 and SecretsManager.4), and AWS WAF (WAF.1).	May 10, 2021
Updates to GuardDuty and Amazon RDS controls (p. 618)	Changed the severity of GuardDuty.1 and PCI.GuardDuty.1 from Medium to High. Added a databaseEngines parameter to RDS.8.	May 4, 2021
Added new resource details to the ASFF	In Resources.Details, added new resource details objects for Amazon EC2 network ACLs, Amazon EC2 subnets, and AWS Elastic Beanstalk environments.	May 3, 2021
Added console fields to provide filter values for Amazon EventBridge rules (p. 618)	The new predefined filter patterns for Security Hub EventBridge rules provide console fields that you can use to specify filter values.	April 30, 2021
Added the integration with AWS Systems Manager Explorer and OpsCenter	Security Hub now supports an integration with Systems Manager Explorer and OpsCenter. The integration receives findings from Security Hub and updates those findings in Security Hub.	April 26, 2021
New type for product integrations (p. 618)	A new integration type, UPDATE_FINDINGS_IN_SECURITY_HUB, indicates that a product integration updates findings that it receives from Security Hub.	April 22, 2021
Changed "master account" to "administrator account" (p. 618)	The term "master account" is changed to "administrator account." The term is also changed in the Security Hub console and API.	April 22, 2021

Updated APIGateway.1 to replace HTTP with WebSocket	Updated the title, description, and remediation for APIGateway.1. The control now checks for WebSocket API execution logging instead of for HTTP API execution logging.	April 9, 2021
Amazon GuardDuty integration now supported in Beijing and Ningxia (p. 618)	The Security Hub integration with GuardDuty is now supported in the China (Beijing) and China (Ningxia) Regions.	April 5, 2021
Added nodejs14.x to the supported runtimes for Lambda.2 control	The Lambda.2 control in the Foundational Security Best Practices standard now supports the nodejs14.x runtime.	March 30, 2021
Security Hub launched in Asia Pacific (Osaka) (p. 618)	Security Hub is now available in the Asia Pacific (Osaka) Region.	March 29, 2021
Added finding provider fields to finding details (p. 618)	On the finding details panel, the new Finding Provider Fields section contains the finding provider values for confidence, criticality, related findings, severity, and types.	March 24, 2021
Added option to receive sensitive findings from Amazon Macie	The integration with Macie can now be configured to send sensitive findings to Security Hub.	March 23, 2021
Added information on making the transition to using AWS Organizations for account management	For customers who have an existing master account with member accounts, added new information on how to change from managing accounts by invitation to managing accounts using Organizations.	March 22, 2021
New objects in ASFF for information about Amazon S3 Public Access Block configuration	In <code>Resources</code> , a new <code>AwsS3AccountPublicAccessBlock</code> resource type and details object provides information about the Amazon S3 Public Access Block configuration for accounts. In the <code>AwsS3Bucket</code> resource details object, the <code>PublicAccessBlockConfiguration</code> object provides the Public Access Block configuration for the S3 bucket.	March 18, 2021

New object in ASFF to allow finding providers to update specific fields	The new <code>FindingProviderFields</code> object in ASFF is used in <code>BatchImportFindings</code> to provide values for <code>Confidence</code> , <code>Criticality</code> , <code>RelatedFindings</code> , <code>Severity</code> , and <code>Types</code> . The original fields should only be updated using <code>BatchUpdateFindings</code> .	March 18, 2021
New <code>DataClassification</code> object for resources in ASFF	The new <code>Resources.DataClassification</code> object in ASFF is used to provide information about sensitive data that was detected on the resource.	March 18, 2021
Added <code>CONFIG_RETURNS_NOT_APPLICABLE</code> value to the available compliance status codes	For the <code>NOT_AVAILABLE</code> compliance status, removed the reason code <code>RESOURCE_NO_LONGER_EXISTS</code> and added the reason code <code>CONFIG_RETURNS_NOT_APPLICABLE</code> .	March 16, 2021
New managed policy for integration with AWS Organizations	A new managed policy, <code>AWSecurityHubOrganizationsAccess</code> , provides the Organizations permissions that are needed by the organization management account and the delegated Security Hub administrator account.	March 15, 2021
Managed policy and service-linked role information moved to the Security chapter	The information on managed policies is revised and expanded. Both the managed policy information and the information on service-linked roles has moved to the Security chapter.	March 15, 2021
New integration with <code>SecureCloudDB</code>	Added <code>SecureCloudDB</code> to the list of third-party integrations. <code>SecureCloudDB</code> is a cloud native database security tool that provides comprehensive visibility of internal and external security postures and activity. <code>SecureCloudDB</code> sends findings to Security Hub.	March 4, 2021
Revised severity for CIS 1.1 and CIS 3.1 – CIS 3.14 controls	The severity of the CIS 1.1 and CIS 3.1 – CIS 3.14 controls is changed to Low.	March 3, 2021
Removed the RDS.11 control (p. 618)	Removed the RDS.11 control from the Foundational Security Best Practices standard.	March 3, 2021

Updated integration for Turbot	The Turbot integration is updated to both send and receive findings.	February 26, 2021
Added controls to the Foundational Security Best Practices standard	Added new controls for Amazon API Gateway (APIGateway.1), Amazon EC2 (EC2.9 and EC2.10), Amazon Elastic File System (EFS.2), Amazon OpenSearch Service (ES.2 and ES.3), Elastic Load Balancing (ELB.6), and AWS Key Management Service (AWS KMS) (KMS.3).	February 11, 2021
Added optional ProductArn filter to the DescribeProducts API	The DescribeProducts API operation now includes an optional ProductArn parameter. The ProductArn parameter is used to identify the specific product integration to return details for.	February 3, 2021
New integration with Antivirus for Amazon S3 from Cloud Storage Security	The integration with Antivirus for Amazon S3 sends the virus scan results to Security Hub as findings.	January 27, 2021
Updated the security score calculation process for master accounts	For a master account, Security Hub uses a separate process to calculate the security score. The new process ensures that the score includes controls that are enabled for member accounts but disabled for the master account.	January 21, 2021
New fields and objects in the ASFF	Added a new Action object to track actions that occurred against a resource. Added fields to the AwsEc2NetworkInterface object to track DNS names and IP addresses. Added a new AwsSsmPatchCompliance object to the resource details.	January 21, 2021
Added controls to the Foundational Security Best Practices standard	Added new controls for Amazon CloudFront (CloudFront.1 through CloudFront.4), Amazon DynamoDB (DynamoDB.1 through DynamoDB.3), Elastic Load Balancing (ELB.3 through ELB.5), Amazon RDS (RDS.9 through RDS.11), Amazon Redshift (Redshift.1 through Redshift.3 and Redshift.6), and Amazon SNS (SNS.1).	January 15, 2021

Workflow status is reset based on the record state or compliance status	Security Hub automatically resets the workflow status from NOTIFIED or RESOLVED to NEW if an archived finding is made active, or if the compliance status of a finding changes from PASSED to either FAILED, WARNING, or NOT_AVAILABLE. These changes indicate that additional investigation is required.	January 7, 2021
Added ProductFields information for control-based findings	For findings that are generated from controls, added information about the content of the ProductFields object in the AWS Security Finding Format (ASFF).	December 29, 2020
Updates to managed insights	Changed the title of insight 5. Added a new insight 32 that checks for IAM users with suspicious activity.	December 22, 2020
Updates to IAM.7 and Lambda.1 controls	In the AWS Foundational Security Best Practices standard, updated the parameters for IAM.7. Updated the title and description of Lambda.1.	December 22, 2020
Expanded integration with ServiceNow ITSM	The ServiceNow ITSM integration allows users to automatically create incidents or problems when a Security Hub finding is received. Updates to these incidents or problems result in updates to the findings in Security Hub.	December 11, 2020
New integration with AWS Audit Manager	Security Hub now offers an integration with AWS Audit Manager. The integration allows Audit Manager to receive control-based findings from Security Hub.	December 8, 2020
New integration with Aqua Security Kube-bench	Security Hub added an integration with Aqua Security Kube-bench. The integration sends findings to Security Hub.	November 24, 2020
Cloud Custodian is now available in the China Regions	The integration with Cloud Custodian is now available in the China (Beijing) and China (Ningxia) Regions.	November 24, 2020

BatchImportFindings can now be used to update additional fields	Previously, you could not use BatchImportFindings to update the Confidence, Criticality, RelatedFindings, Severity, and Types fields. Now, if these fields have not been updated by BatchUpdateFindings, they can be updated by BatchImportFindings. Once they are updated by BatchUpdateFindings, they cannot be updated by BatchImportFindings.	November 24, 2020
Security Hub is now integrated with AWS Organizations	Customers can now manage member accounts using their Organizations account configuration. The organization management account designates the Security Hub administrator account, who determines which organization accounts to enable in Security Hub. The manual invitation process can still be used for accounts that are not part of an organization.	November 23, 2020
Removed the separate finding list format for high-volume controls (p. 618)	The finding list for a control no longer uses the Findings page format when there is a very large number of findings.	November 19, 2020
New and updated third-party integrations	Security Hub now supports integrations with cloudtamer.io, 3CORESec, Prowler, and StackRox Kubernetes Security. IBM QRadar no longer sends findings. It only receives findings.	October 30, 2020
Added option to download the list of findings from the control details page.	On the control details page, a new Download option allows you to download the finding list to a .csv file. The downloaded list respects any filters that are on the list. If you selected specific findings, then the downloaded list only includes those findings.	October 26, 2020

Added option to download the list of controls from the standard details page.	On the standard details page, a new Download option allows you to download the control list to a .csv file. The downloaded list respects any filters that are on the list. If you selected a specific control, then the downloaded list only includes that control.	October 26, 2020
New and updated partner integrations	Security Hub is now integrated with ThreatModeler. Updated the following partner integrations to reflect their new product names. Twistlock Enterprise Edition is now Palo Alto Networks - Prisma Cloud Compute. Also from Palo Alto Networks, Demisto is now Cortex XSOAR and Redlock is now Prisma Cloud Enterprise.	October 23, 2020
Security Hub launched in China (Beijing) and China (Ningxia) (p. 618)	Security Hub is now available in the China (Beijing) and China (Ningxia) Regions.	October 21, 2020
Revised format for ASFF attributes and third-party integrations (p. 618)	The lists of ASFF attributes and partner integrations now use a list-based format instead of tables. The ASFF syntax, attributes, and types taxonomy are now in separate topics.	October 15, 2020
Redesigned standard details page	The standard details page for an enabled standard now displays a tabbed list of controls. The tabs filter the control list based on the control status.	October 7, 2020
Replaced CloudWatch Events with EventBridge (p. 618)	Replaced references to Amazon CloudWatch Events with Amazon EventBridge.	October 1, 2020
New integrations with Blue Hexagon for AWS, Alcide kAudit, and Palo Alto Networks VM-Series.	Security Hub is now integrated with Blue Hexagon for AWS, Alcide kAudit, and Palo Alto Networks VM-Series. Blue Hexagon for AWS and kAudit send findings to Security Hub. VM-Series receives findings from Security Hub.	September 30, 2020

New and updated resource details objects in ASFF	<p>Added new Resources.Details objects for <code>AwsApiGatewayRestApi</code>, <code>AwsApiGatewayStage</code>, <code>AwsApiGatewayV2Api</code>, <code>AwsApiGatewayV2Stage</code>, <code>AwsCertificateManagerCertificate</code>, <code>AwsElbLoadBalancer</code>, <code>AwsIamGroup</code>, and <code>AwsRedshiftCluster</code>. Added details to the <code>AwsCloudFrontDistribution</code>, <code>AwsIamRole</code> and <code>AwsIamAccessKey</code> objects.</p>	September 30, 2020
New <code>ResourceRole</code> attribute for resources in ASFF to track whether a resource is an actor or a target.	<p>The <code>ResourceRole</code> attribute for resources indicates whether the resource is the target of the finding activity or the perpetrator of the finding activity. The valid values are <code>ACTOR</code> and <code>TARGET</code>.</p>	September 30, 2020
Added AWS Systems Manager Patch Manager to available AWS service integrations	<p>AWS Systems Manager Patch Manager is now integrated with Security Hub. Patch Manager sends findings to Security Hub when instances in a customer's fleet go out of compliance with their patch compliance standard.</p>	September 22, 2020
Added new controls to the Foundational Security Best Practices standard	<p>Added new controls for the following services: Amazon EC2 (EC2.7 and EC2.8), Amazon EMR (EMR.1), IAM (IAM.8), Amazon RDS (RDS.4 through RDS.8), Amazon S3 (S3.6), and AWS Secrets Manager (SecretsManager.1 and SecretsManager.2).</p>	September 15, 2020
New context keys for IAM policy to control access to <code>BatchUpdateFindings</code> fields	<p>IAM policies can now be configured to restrict access to fields and field values when using <code>BatchUpdateFindings</code>.</p>	September 10, 2020
Expanded access to <code>BatchUpdateFindings</code> for member accounts	<p>By default, member accounts now have the same access to <code>BatchUpdateFindings</code> as master accounts.</p>	September 10, 2020
New controls for AWS KMS in the Foundational Security Best Practices Standard	<p>Added two new controls (KMS.1 and KMS.2) to the Foundational Security Best Practices Standard. The new controls check whether IAM policies restrict access to AWS KMS decryption actions.</p>	September 9, 2020

Removed account-level findings for controls (p. 618)	Security Hub no longer generates account-level findings for a control. Only resource-level findings are generated.	September 1, 2020
New PatchSummary object in ASFF	Added the <code>PatchSummary</code> object to the ASFF. The <code>PatchSummary</code> object provides information about the patch compliance of a resource relative to a selected compliance standard.	September 1, 2020
Redesigned control details page	The details page for controls is redesigned. The control finding list provides tabs to allow you to quickly filter the list based on the compliance status. You can also quickly see suppressed findings. Each entry provides access to additional details about the finding resource, AWS Config rule, and finding notes.	August 28, 2020
New filter options for findings	For finding filters, you can use the is not filter to find findings for which a field value is not equal to the filter value. You can use the does not start with to find findings for which a field value does not start with the specified filter value.	August 28, 2020
New resource details objects in ASFF	Added new <code>Resources.Details</code> objects for the following resource types: <code>AwsDynamoDbTable</code> , <code>AwsEc2Eip</code> , <code>AwsIamPolicy</code> , <code>AwsIamUser</code> , <code>AwsRdsDbCluster</code> , <code>AwsRdsDbClusterSnapshot</code> , <code>AwsRdsDbSnapshot</code> , <code>AwsSecretsManagerSecret</code>	August 18, 2020
New integration with RSA Archer	Security Hub is now integrated with RSA Archer. RSA Archer receives findings from Security Hub.	August 18, 2020
New Description field for AwsKmsKey	Added a <code>Description</code> field to the <code>AwsKmsKey</code> object under <code>Resources.Details</code> .	August 18, 2020
Added fields to AwsRdsDbInstance	Added several attributes to the <code>AwsRdsDbInstance</code> object under <code>Resources.Details</code> .	August 18, 2020

Updated how Security Hub determines the overall status of a control	For controls that have no findings, the status is No data instead of Unknown . The control status includes both account-level and resource-level findings. The control status does not use the workflow status of findings, except to ignore suppressed findings.	August 13, 2020
Updated how Security Hub calculates the security score for a standard	When calculating the security score for a standard, Security Hub now ignores controls with a status of No Data . The security score is proportion of passed controls to enabled controls, excluding controls with no data.	August 13, 2020
New option to automatically enable new controls in enabled standards	Added a Settings option to automatically enable new controls in standards that are enabled. You can also use the <code>UpdateSecurityHubConfiguration</code> API operation to configure this option.	July 31, 2020
New controls for the Payment Card Industry Data Security Standard (PCI DSS) standard	Added new controls to the PCI DSS standard. The identifiers of the new controls are <code>PCI.DMS.1</code> , <code>PCI.EC2.5</code> , <code>PCI.EC2.6</code> , <code>PCI.ELBV2.1</code> , <code>PCI.GuardDuty.1</code> , <code>PCI.IAM.7</code> , <code>PCI.IAM.8</code> , <code>PCI.S3.5</code> , <code>PCI.S3.6</code> , <code>PCI.SageMaker.1</code> , <code>PCI.SSM.2</code> , and <code>PCI.SSM.3</code> .	July 29, 2020
New and updated controls for the Foundational Security Best Practices standard	Added new controls to the Foundational Security Best Practices standard. The identifiers of the new controls are <code>AutoScaling.1</code> , <code>DMS.1</code> , <code>EC2.4</code> , <code>EC2.6</code> , <code>S3.5</code> , and <code>SSM.3</code> . Updated the title of <code>ACM.1</code> and changed the value of the <code>daysToExpiration</code> parameter to 30.	July 29, 2020
New <code>Vulnerabilities</code> object in the ASFF	Added the <code>Vulnerabilities</code> object, which provides information about vulnerabilities that are associated with the finding.	July 1, 2020
New <code>Resource.Details</code> objects in the ASFF for <code>AutoScaling</code> groups, <code>EC2</code> volumes, and <code>EC2 VPCs</code>	Added the <code>AwsAutoScalingAutoScalingGroup</code> , <code>AWSEC2Volume</code> , and <code>AwsEC2Vpc</code> objects to <code>Resource.Details</code> .	July 1, 2020

New NetworkPath object in the ASFF	Added the <code>NetworkPath</code> object, which provides information about a network path that is related to the finding.	July 1, 2020
Automatically resolve findings when <code>Compliance.Status</code> is <code>PASSED</code>	For findings from controls, if <code>Compliance.Status</code> is <code>PASSED</code> , then Security Hub automatically sets <code>Workflow.Status</code> to <code>RESOLVED</code> .	June 24, 2020
AWS Command Line Interface examples (p. 618)	Added AWS CLI syntax and examples for several Security Hub tasks. Includes enabling Security Hub, managing insights, managing standards and controls, managing product integrations, and disabling Security Hub.	June 24, 2020
New <code>Severity.Original</code> attribute in the ASFF	Added the <code>Severity.Original</code> attribute, which is the original severity from the finding provider. This replaces the deprecated <code>Severity.Product</code> attribute.	May 20, 2020
New <code>Compliance.StatusReasons</code> object in the ASFF for details about a control's status	Added the <code>Compliance.StatusReasons</code> object, which provides additional context for the current status of a control.	May 20, 2020
New AWS Foundational Security Best Practices standard	Added the new AWS Foundational Security Best Practices standard, which is a set of controls that detect when your deployed accounts and resources deviate from security best practices.	April 22, 2020
New console option to update the workflow status for a finding	Added information for using the Security Hub console or API to set the workflow status for findings.	April 16, 2020
New <code>BatchUpdateFindings</code> API for customer updates to findings	Added information on using <code>BatchUpdateFindings</code> to update information related to the process of investigating a finding. <code>BatchUpdateFindings</code> replaces <code>UpdateFindings</code> , which is deprecated.	April 16, 2020

Updates to the AWS Security Finding Format (ASFF)	Added several new resource types. Added a new <code>Label</code> attribute to the <code>Severity</code> object. <code>Label</code> is intended to replace the <code>Normalized</code> field. Added a new <code>Workflow</code> object to track the process of an investigation into a finding. <code>Workflow</code> contains a <code>Status</code> attribute, which replaces the existing <code>Workflowstate</code> attribute.	March 12, 2020
Updates to the Integrations page	Updated to reflect the changes to the Integrations page. For each integration, the page now shows the integration category and whether each integration sends findings to or receives findings from Security Hub. It also provides the specific steps required to enable each integration.	February 26, 2020
New third-party product integrations	Added the following new product integrations: Cloud Custodian, FireEye Helix, Forcepoint CASB, Forcepoint DLP, Forcepoint NGFW, Rackspace Cloud Native Security, and Vectra.ai Cognito Detect.	February 21, 2020
New security standard for the Payment Card Industry Data Security Standard (PCI DSS)	Added the Security Hub security standard for the Payment Card Industry Data Security Standard (PCI DSS). When this standard is enabled, Security Hub performs automated checks against controls related to PCI DSS requirements.	February 13, 2020
Updates to the AWS Security Finding Format (ASFF)	Added a field for related requirements for standards controls . Added new resource types and new resource details . The ASFF also now allows you to provide up to 32 resources.	February 5, 2020
New option to disable individual security standard controls	Added information on how to control whether each individual security standard control is enabled.	January 15, 2020
Updates to Terminology and Concepts	Updated some descriptions and added new terms to Terminology and Concepts .	September 21, 2019

AWS Security Hub general availability release (p. 618)	Content updates to reflect improvements made to Security Hub during the preview period.	June 25, 2019
Added remediation steps for CIS AWS Foundations checks	Added remediation steps to Security Standards Supported in AWS Security Hub .	April 15, 2019
Preview release of AWS Security Hub (p. 618)	Published the preview release version of the <i>AWS Security Hub User Guide</i> .	November 18, 2018