Check Point
SOFTWARE TECHNOLOGIES LTD.
WELCOME TO THE FUTURE OF CYBER SECURITY

Scaling Identity based Access Control

About this guide

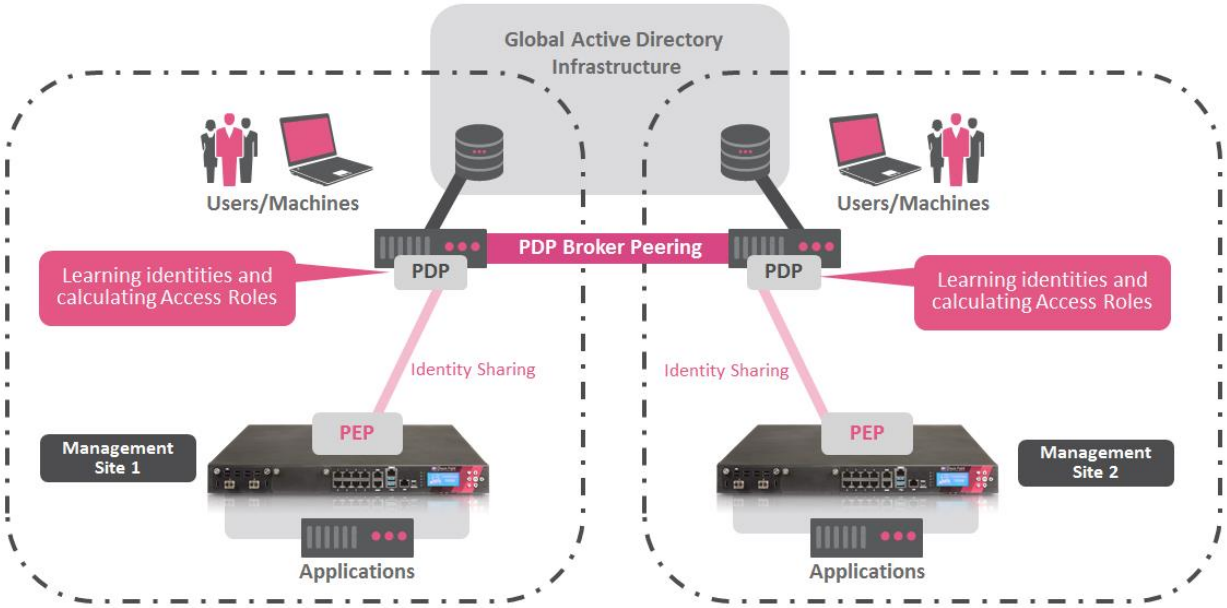
You will learn about the architecture scaling identity based access control across multiple sites. Identity based access control is providing roaming users access to applications independent of the network they are connected to or the site they are located in. Sharing these identities presents the challenge. The functionality PDP Broker described in the document is solving this challenge.

The PDP Broker is sharing identities across management domains and across geographical areas.

"PDP Broker Peering works like a network routing functionality sharing identities."

Introduction

Users and machines are represented as Access Role objects in the security policy. Once users have logged on to the network the login event is learned by the PDP (Policy Decision Point), the matching Access Role is calculated and an identity session is created. This identity session is shared with peering PDP Broker nodes and security gateways running the identity based enforcement instance called PEP (Policy Enforcement Point). The user will get access to the applications based on these identity sessions.



Version 7, January 2019
Peter Elmer

Table of content

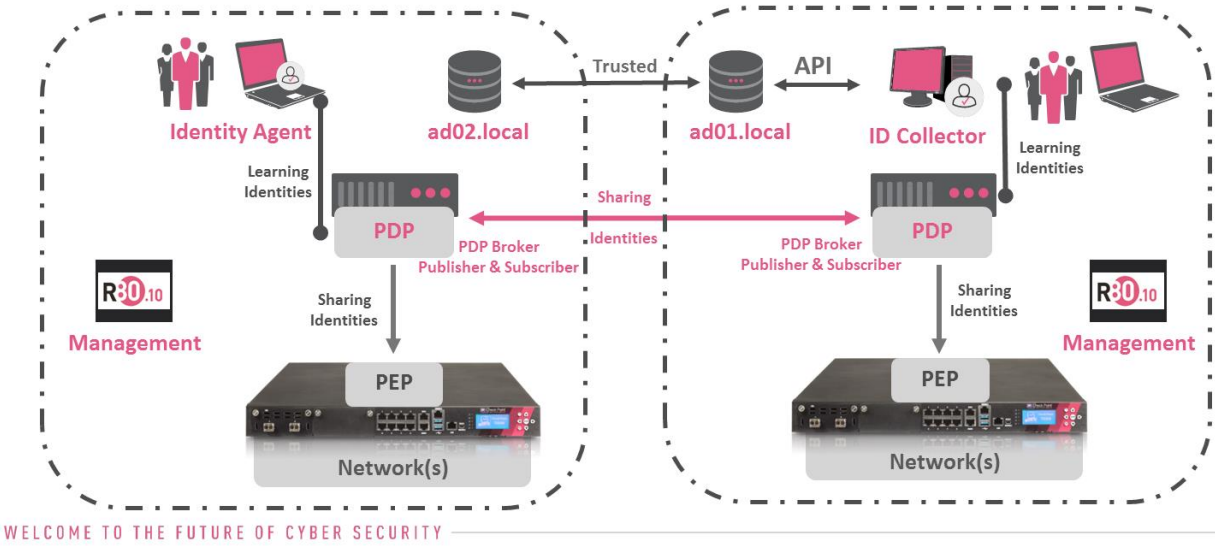
Overview of the PDP Broker functionality

The PDP Broker functionality resolves the challenge sharing identities in large scale environments. Identity sharing can be achieved across management domains and across geographical or organizational realms.

This version of PDP Broker requires a gateway running R80.10 + JHF 112 and the PDP Broker HF. All configurations are done using a file that is read when installing the policy. As the PDP Broker HF is bound to a dedicated JHF it is recommended installing it on a dedicated machine allowing the security enforcing gateways running the latest GA JHF.

The PDP Broker HF is distributed by Check Point Solution Center in response to an RFE ticket.

The PDP Broker includes two functionalities: Publisher and Subscriber. The PDP Broker Publisher is the instance initiating an HTTPS connection to the PDP Broker Subscriber using the Identity Awareness API as underlying infrastructure. The functionality has been created in addition to the so called "Multi-SIC" function documented in sk65404 allowing the sharing identities across management domains from PDP to PEP instances. The PDP Broker functionality is allowing the sharing of identities between PDP instances managed by the same or different management domains or SmartCenter servers.



In the diagram above you see that identities are learned by the PDP process using the ID Agent and the ID Collector. In addition methods such as Captive Portal, Radius Accounting and the integration to Cisco ISE centric networks are supported. A full list of supported methods is documented in the Identity Awareness Administration guide. Once the PDP has learned the identity it calculates the Access Role object matching this identity and shares the information with PEP instances running on Security Gateways and PDP Broker instances configured as Subscriber.

Some details about PDP Broker identity sharing

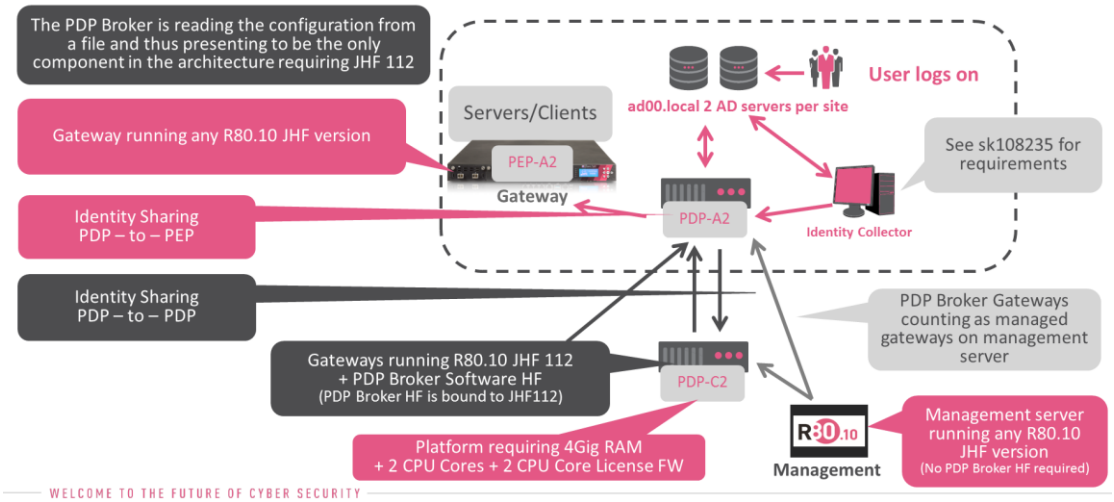
When establishing the peering connection the PDP Broker Publisher verifies the Subject Name of the Gaia Multi-Portal HTTPS certificate the PDP Broker Subscriber presents in response to the HTTPS connection request. In case the certificate validation is successful identity updates are sent to the PDP Broker peer in form of HTTP POST requests using the Identity Awareness API.

The identity updates are representing the identities learned. **It is important to understand that the Access Role matching the learned identities is calculated before the information is shared with PDP Broker peers,** just in the same way like they are calculated before being shared with PEP instances running on gateways. Operating your network following this principle requires **Access Role objects being consistent (they need to have the same names) across management domains.** The security rule base is required using the same Access Role objects (with same names) on the relevant gateways.

Optionally you can configure a PDP Broker Subscriber to re-calculate the Access Role object once it has received the identity. This method is allowing using Access Role objects having different names in each management domain. Please contact Check Point Sales Engineering team for details.

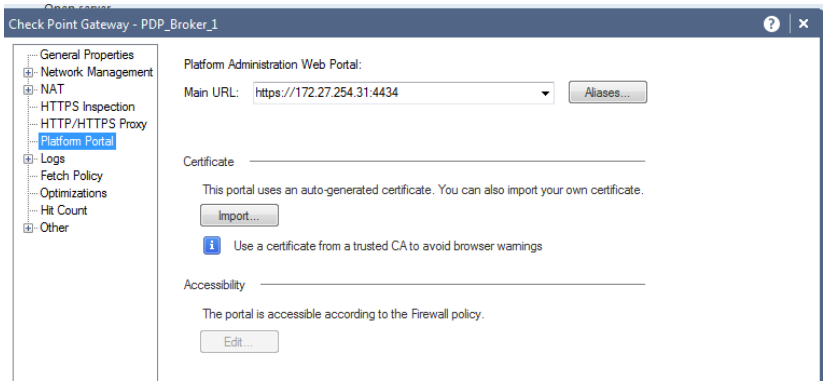
Preparing the a dedicated PDP Broker gateway

From an architecture point of view you may prefer having a dedicated gateway acting as PDP Broker as this function is currently tied to JHF 112 and a gateway acting as PDP Broker can´t be updated to a later version.



Perform the following steps on the gateway foreseen to become the PDP Broker.

- Perform a fresh install of Check_Point_R80.10_T462_Gaia
- Run the first time wizard and install it as a gateway
- Install the latest CPUSE (build 1567 or later)
- Install the JHF 112
- Add a gateway objects for the PDP Broker gateways to the management server
    - Enable FW Blade only
    - Make sure the interface topology and anti-spoofing are correctly defined
    - Change the Gaia Platform Portal port to another value other than the default TCP/443 avoiding issues with ID Awareness authentication processes



Once completed install a default security policy on the gateway and verify the connectivity.

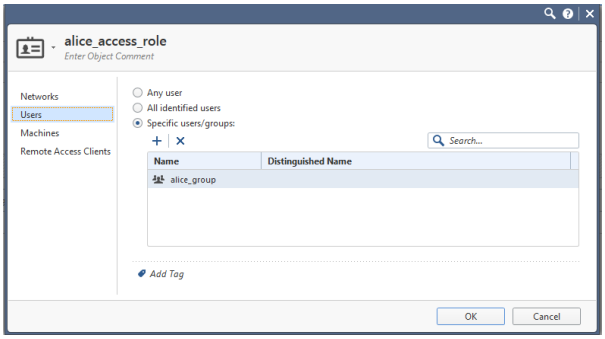
Important notes when installing clustered gateways on VMware

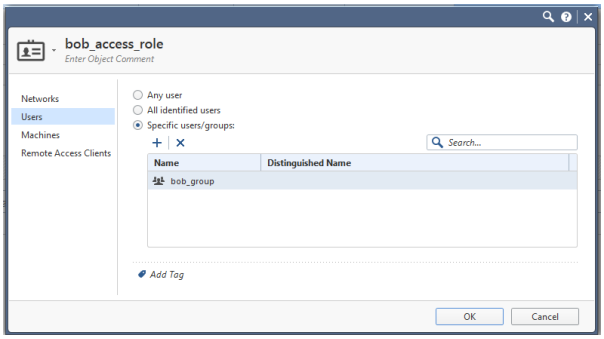
In case the dedicated PDP Broker instances are installed as a cluster on VMware ESXi you want to keep the following in mind.

- Use the CloudGuard (vSEC) Security Gateway Network Mode OVF template (download link)
- Enable the VPN Blade on the cluster object to create a unique default certificate for the multi-portal for both cluster members, install policy, then disable the VPN Blade and install policy again
- Verify the cluster is working properly using "`cphaprob stat`" command
- You may need to switch the cluster CCP protocol to broadcast "`cphaconf set_ccp broadcast`" (see sk20576 for details)
- You may need to modify the VMware vswitch configuration allowing "Forged Transmits"
    - See sk101214 and this vmware article
- Don´t use the "vmac" functionality on the cluster object

Configure a source to learn identities from and create Access Role objects

In this example we use the Captive Portal to learn identities. We will create dummy users "Alice" and "Bob" who will be associated to User Groups "Alice Group" and "Bob Group". These group objects will then be referenced in the Access Role objects "Alice Access Role" and "Bob Access Role".





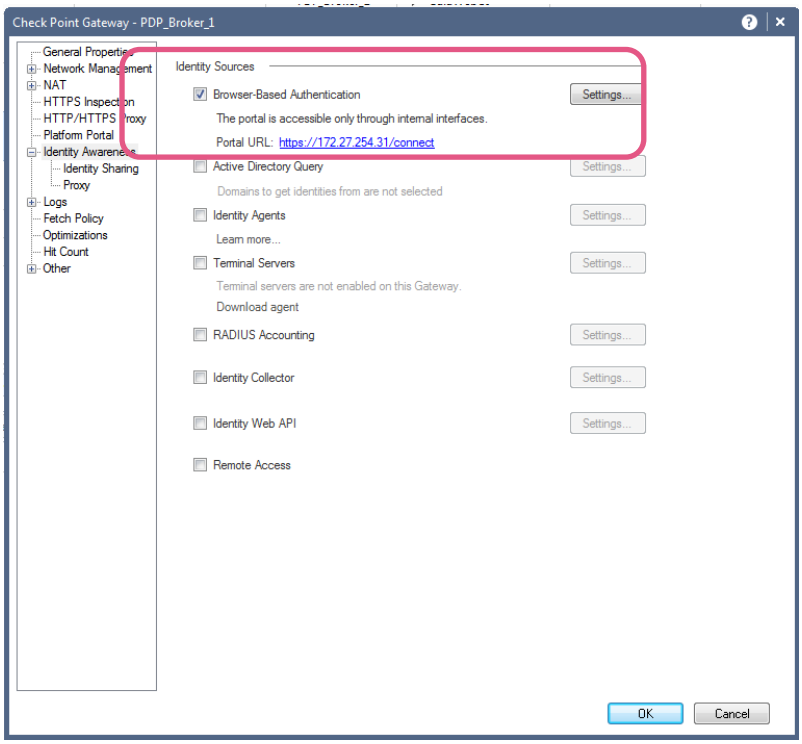
Include and Access Rule based on Access Roles in your Security Policy

This is a simple rule base allowing the two PDP Broker instances to exchange identities learned from the Captive Portal and ID Agent.

No.	Hits	Name	Source	Destination	Services & Applications	Action	Track
1	88	Management	net_192.168.169.0	PDP_Broker_1 PDP_Broker_2	ssh GaiaWebUI	Accept	None
2	0	Name Service	net_192.168.169.0 net_192.168.170.0	Any	dns	Accept	None
3	1	ICMP comment	net_192.168.169.0 net_192.168.170.0	Any	icmp-proto	Accept	None
4	3	Access Captive Portal	Host_192.168.169.1	PDP_Broker_1 PDP_Broker_2	https	Accept	Log
5	4	PDP Broker Peering	PDP_Broker_2 PDP_Broker_1	PDP_Broker_1 PDP_Broker_2	https	Accept	Log
6	0	ID based access Alice	alice_access_role	net_172.27.254.0	http dns	Accept	Log
7	0	ID based access Bob	bob_access_role	net_172.27.254.0	http dns	Accept	Log
8	24K	Clean NBT	Any	Any	NBT	Drop	None
9	97	Cleanup rule	Any	Any	Any	Drop	Log

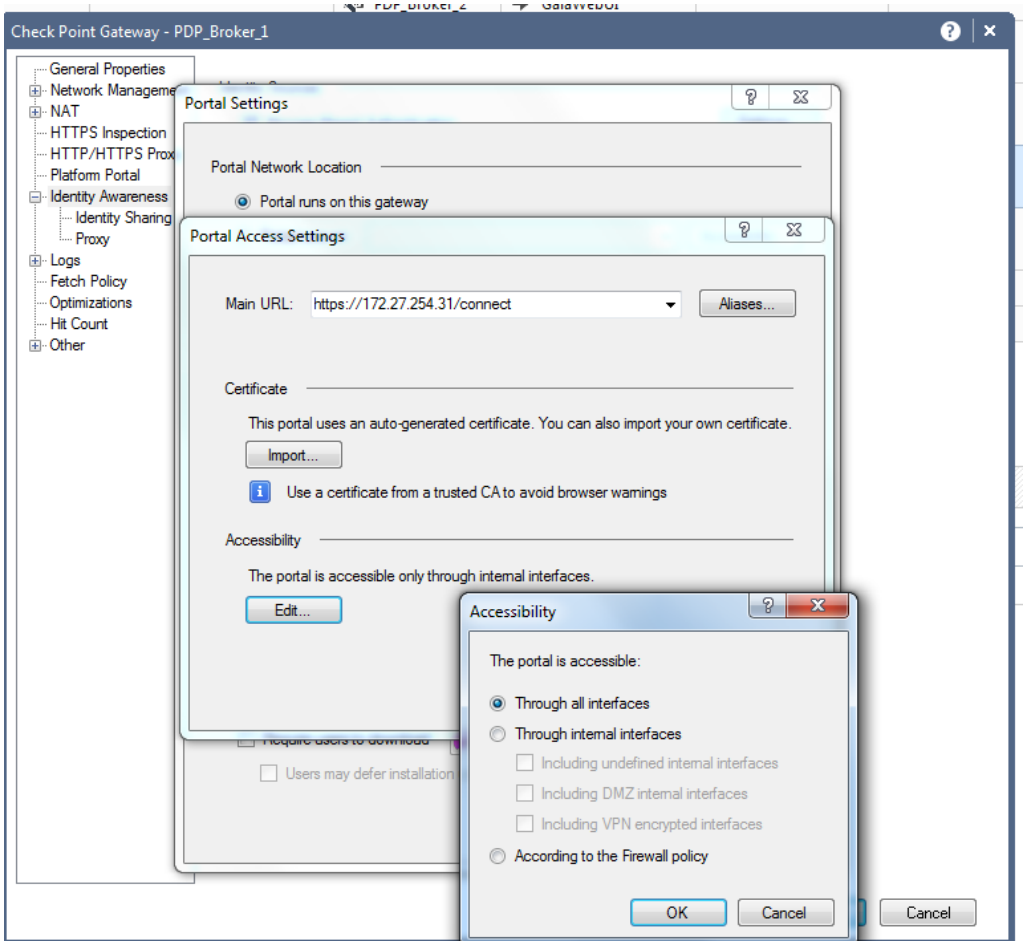
Enabling the Identity Awareness Blade on the gateways

Enable ID Awareness on the gateways but cancel the “wizard”. Perform a manual configuration enabling the “Browser based authentication”. Disable “Remote Access”.



Disable “Share identities with other gateways” for the initial test.

Make sure the Platform Portal IP Address is reachable from your test computer. You may need to allow access to the portal “Through all interfaces”.



General configurations settings worth being reviewed

The following guidelines may or may not be necessary to be applied as they depend on the environment you are integrating into. It is recommended having a dialog with the team administrating the Active Directory environment and to agree on a common strategy for the integration process.

Nested Groups

In case nested groups are used it is recommended reviewing the LDAP query type. The PDP queries against the Active Directory are by default based on one query per group. If nested groups are used it is recommended applying the "per-user nested group" or the "multiple per-user nested group setting as documented in the Identity Awareness Administration Guide (link to relevant chapter). Note the command is composed of two "_" (underscore) characters in the "__set state" section example:

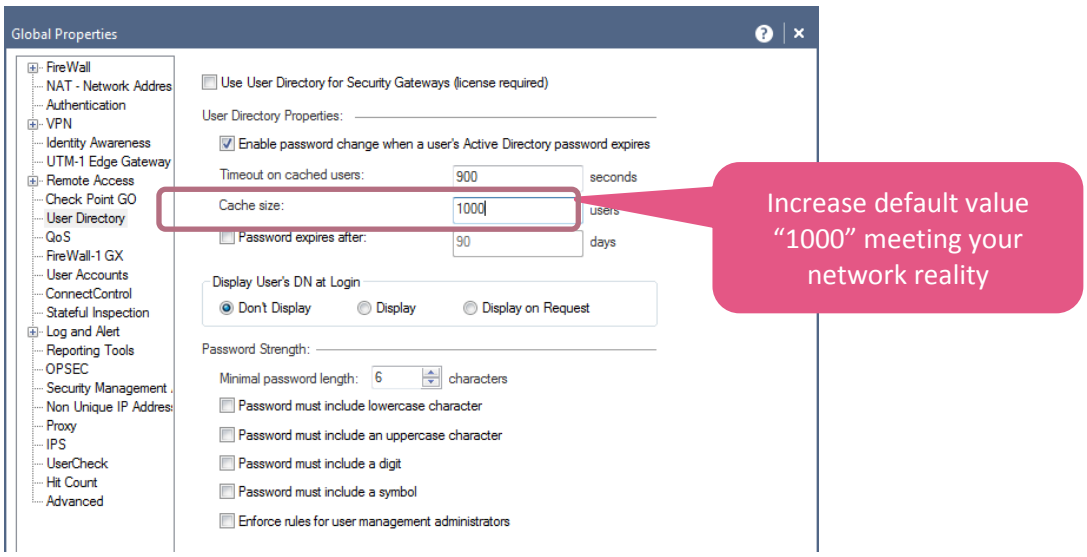
`"pdp nested_groups __set_state 3"`

Kerberos ticket size

In today's Active Directory environments you often find Kerberos tickets exceeding the default size configured on the Check Point Management Server. You may want to apply guidelines documented in sk66087 to increase the default Kerberos ticket size. Using the maximum value of 65535 is not an issue for most environments. The sk66087 documents how to change the "`ccc_max_msg_size`" value using GUIDBedit. Keep in mind that before you start GUIDBedit you need to close all SmartConsole applications and publish all ongoing sessions before. Once you have changed the value, save the changes in GUIDBedit, open a SmartConsole and install the access security policy on the gateways where the Identity Awareness Blade is enabled.

User Cache Size

Each gateway maintains a cache storing Identity Awareness related information. The size of this cache shall match the number of users and their nested group level. It is recommended setting the "User Cache" to a value meeting the "number of user" multiplied with the nested "group depth level". Example: A network with 3000 users and each user is associated with three nested groups



Browser Based Authentication

When using Browser Based Authentication and Single Sign-On based for Identity Agents based on Kerberos tickets make sure the IP address of the security gateway running the portal can be resolved using a DNS name and that the certificate reflects this name in the Subject Field.

Terminal Server

Download the latest Terminal Server Agent from sk134312.

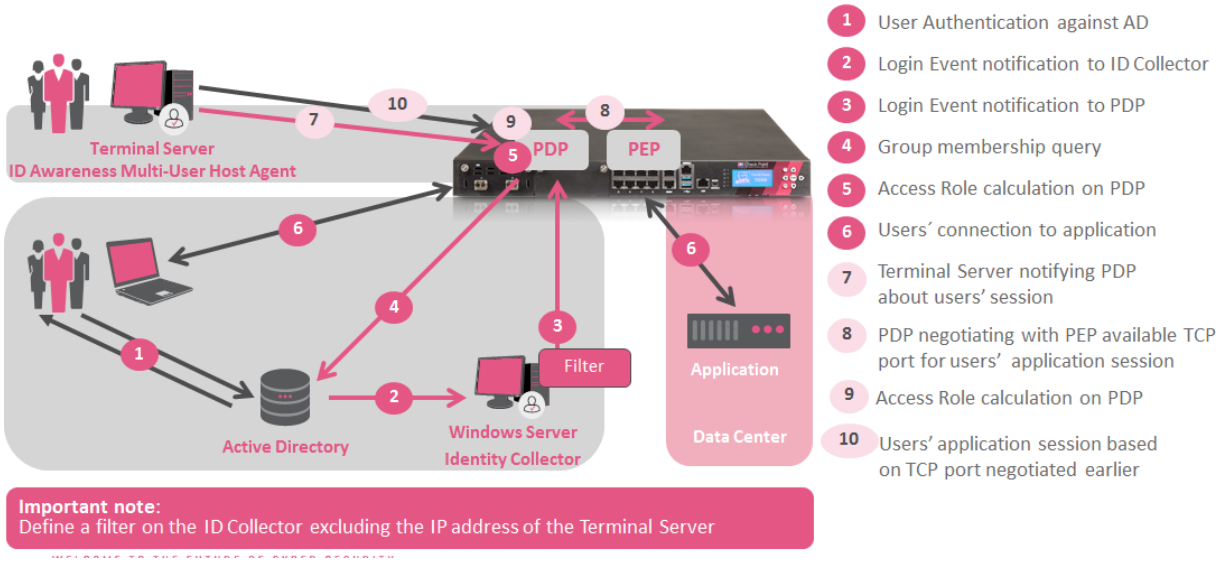
When using the Terminal Server Multi-User Host Agent the configuration of Kerberos Single Sign-On is required. The TS MUH agent tries authenticating the machine it is running on against the PDP process it is connecting to (you see the TS Agent UI showing "Authenticating" state). In case Kerberos is not configured on the gateway running PDP process the MUH agent user interface will flip between "Authenticating" and "Disconnected" state. On the PDP you will see the command "`pdp connections ts`" showing the MUH being connected. In addition you may even see identities learned but the UI of MUH will continue to change between the two states.

Using Terminal Server and Identity Collector as identity sources

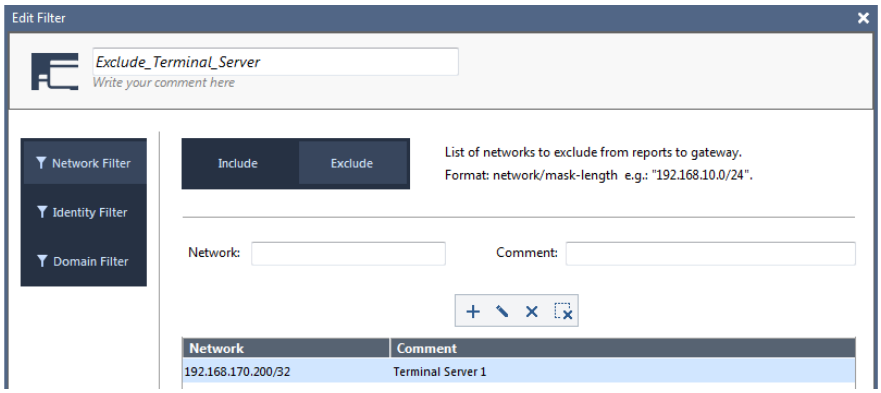
The Identity Collector learns login events using the Microsoft API (see Identity Awareness Administration Guide for details). Download the latest Identity Collector version from sk134312.

The Windows server running the Identity Awareness Terminal Server Agent may be part of the Active Directory. In consequence all users authentication to this server (including administrative and service accounts) will generate a login event to the Active Directory environment. These login events will be forwarded using the Microsoft API to the Identity Collector. The ID Collector will forward these events to the PDP.

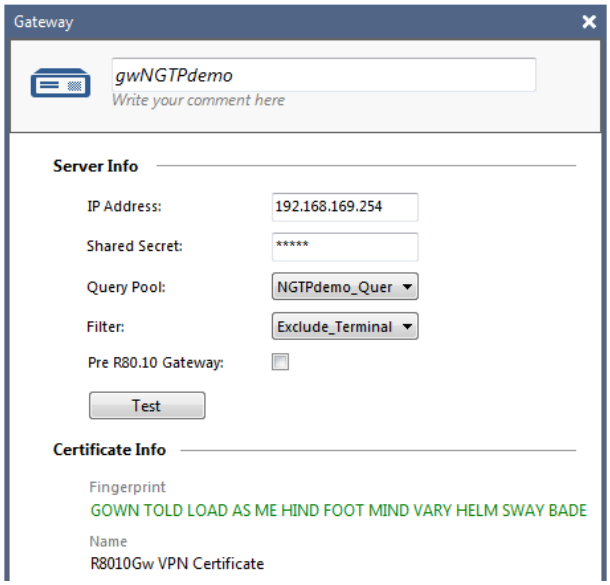
You need a filter on the ID Collector excluding the IP address of the Terminal Server to avoid issues.



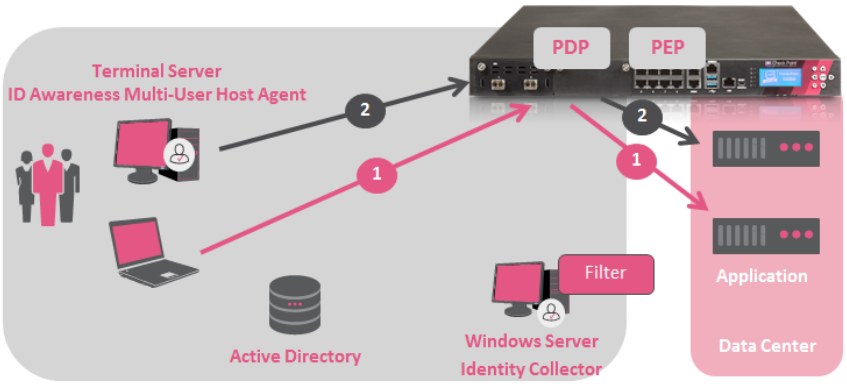
Define the exclude filter on the ID Collector.



Assign it to the gateway object.



The solution scenario you can achieve looks like shown below.



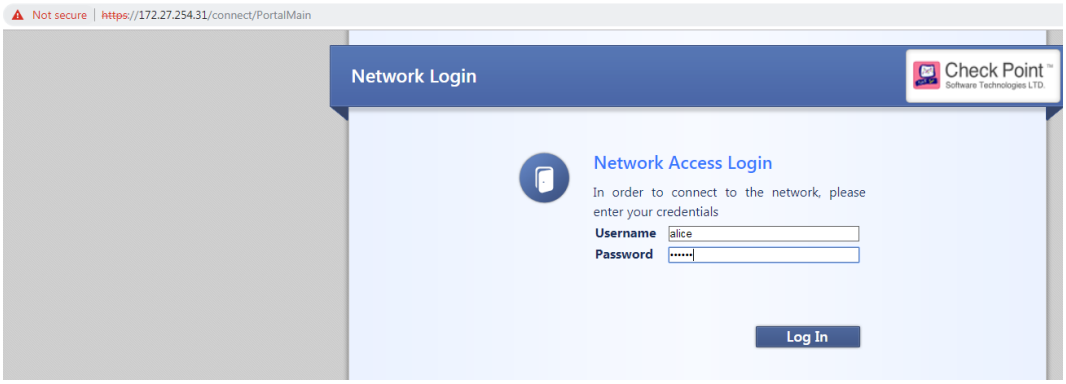
- Thanks to the filter correctly applied on the ID collector the same user can have two application sessions:
- #1 using the identity session based on the ID Collector learned login event
- #2 using the identity session established with the help of MUH Agent on the Terminal Server

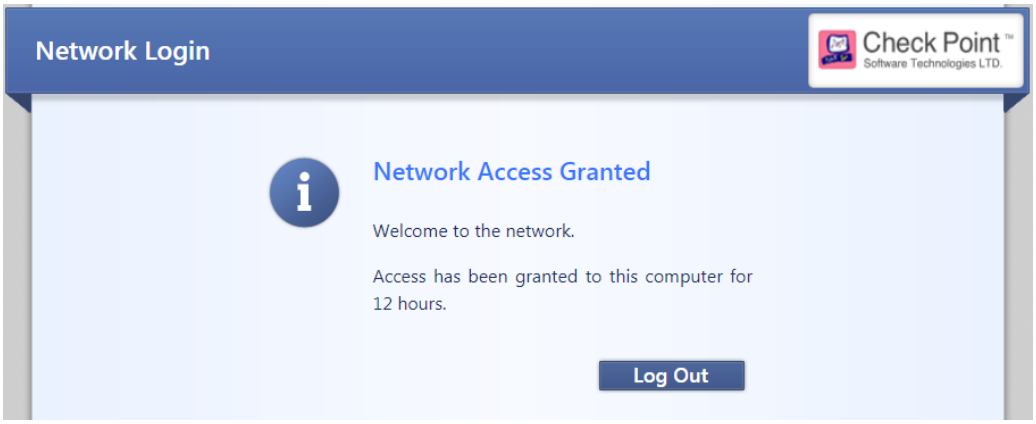
Important note:
Define a filter on the ID Collector excluding the IP address of the Terminal Server

Verifying the basic ID Awareness functionality

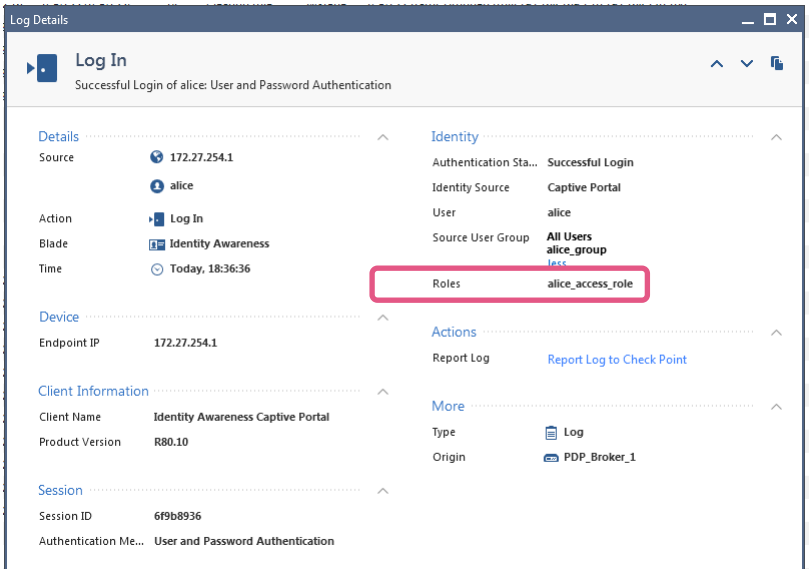
Install the access control policy shown above on both PDP Broker gateways.

Test access of user “Alice” on PDP_Broker_1





Make sure you see a log message where the correct Access Role “alice group” is associated.

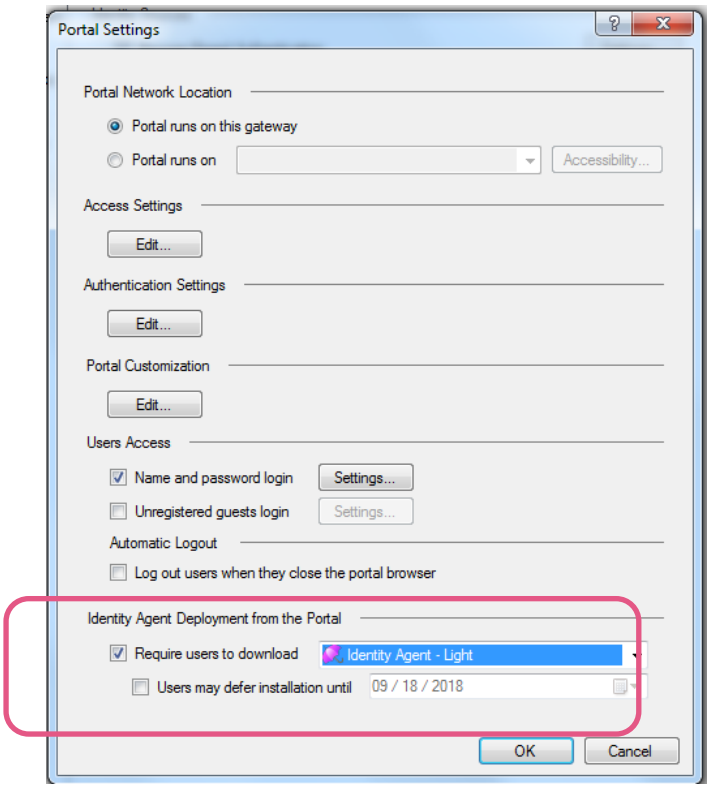


Verify the same for user “Bob” on PDP_Broker_2.

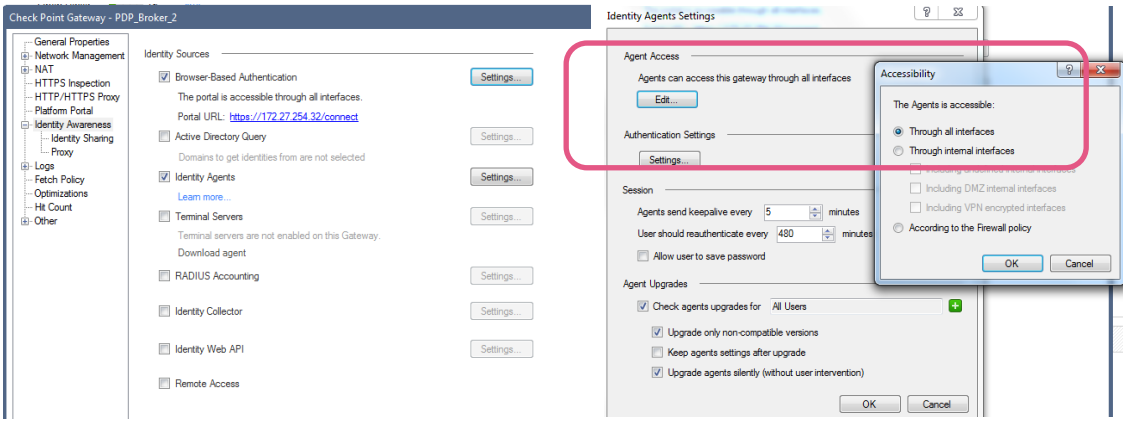
Verifying ID Agent functionality

You may optionally verify that the ID Agent can connect to a PDP Broker instance. Here we enable ID Agent support on PDP Broker 2.

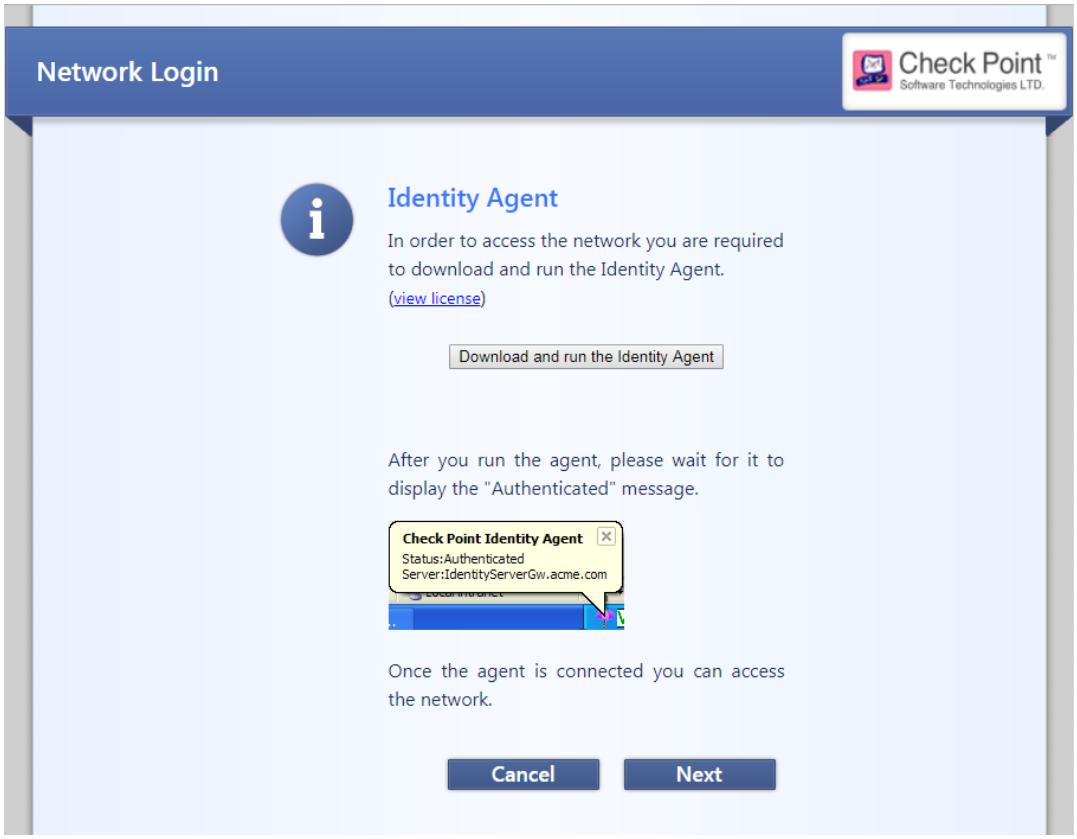
Enable download option in the Portal Settings:



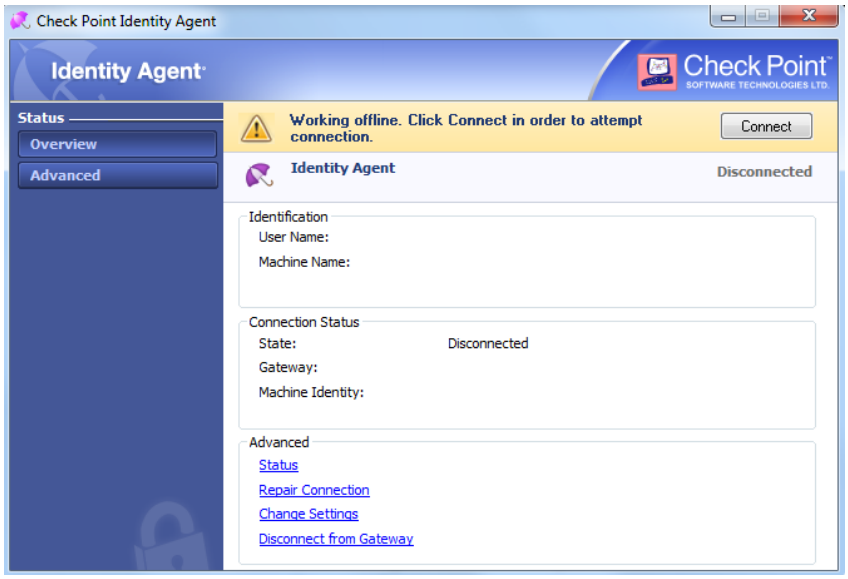
Make sure access for ID Agent can be “Through all interfaces”



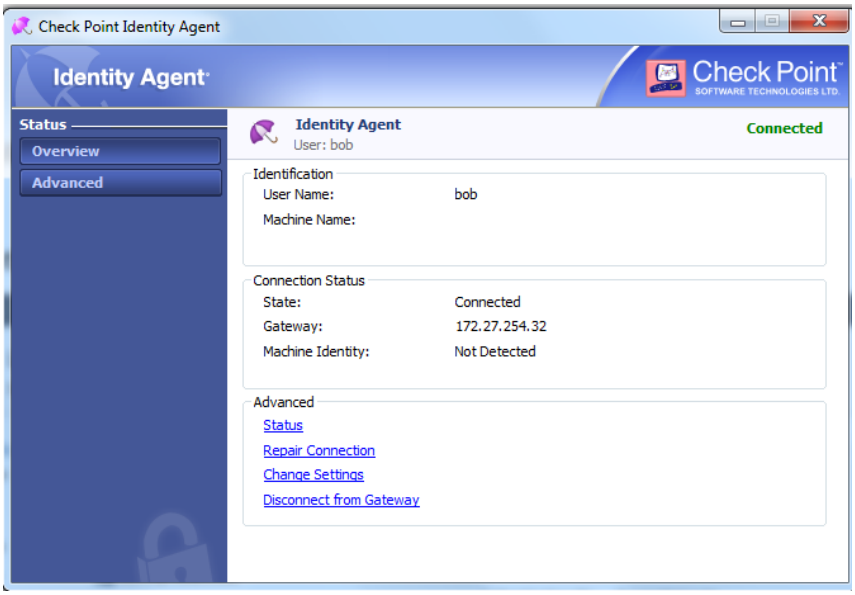
Connect to the Captive Portal, authenticate and download the ID Agent.



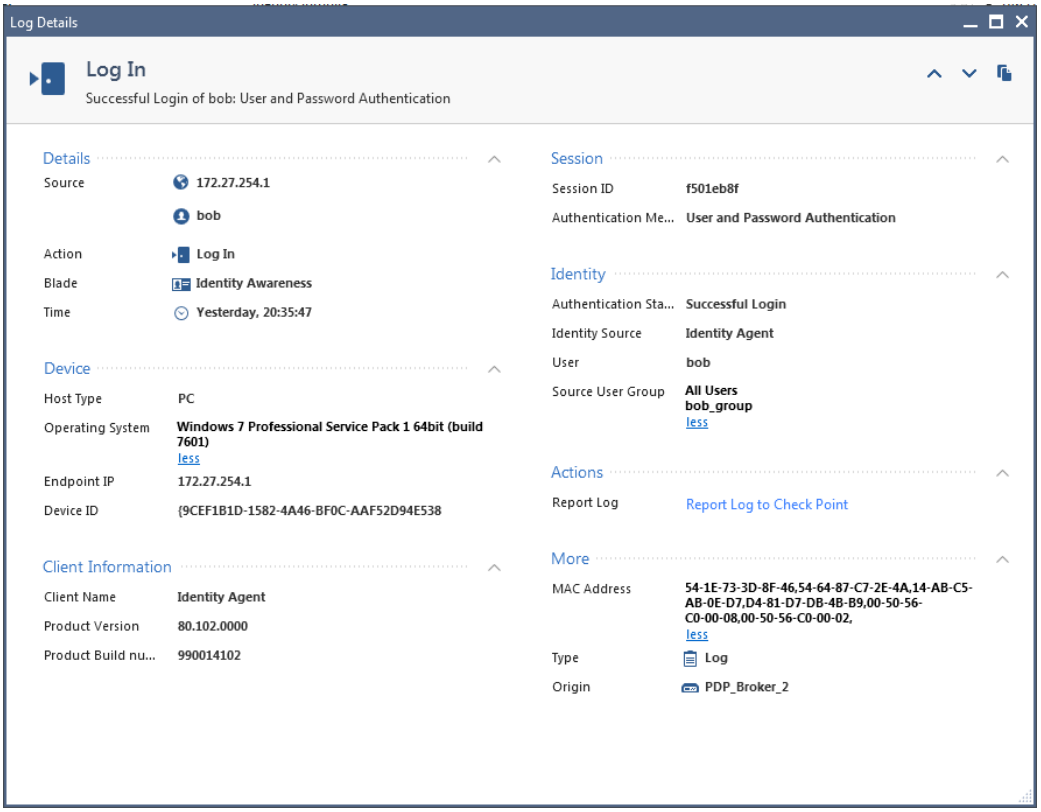
Once downloaded install the agent and connect to the gateway.



The ID Agent will connect.



See the relevant log messages in SmartLog.



Installation of the PDP Broker Hotfix

Installing on clustered gateways running PDP and PEP

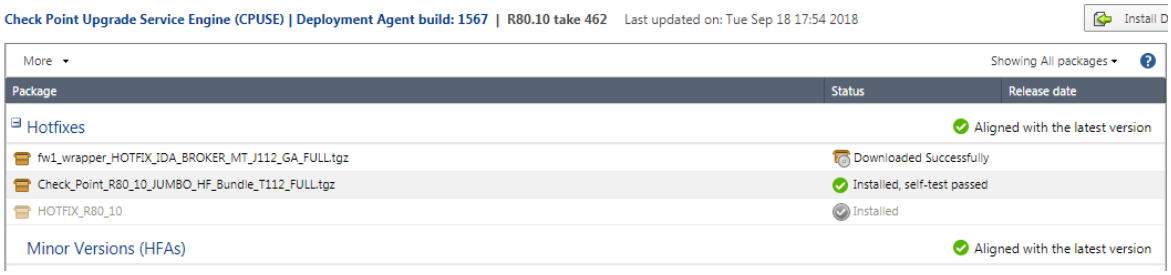
If you are planning to install the Hotfix on a cluster maintaining active identity sessions, please follow all the steps outlined below.

1. Disable Identity Awareness Blade.

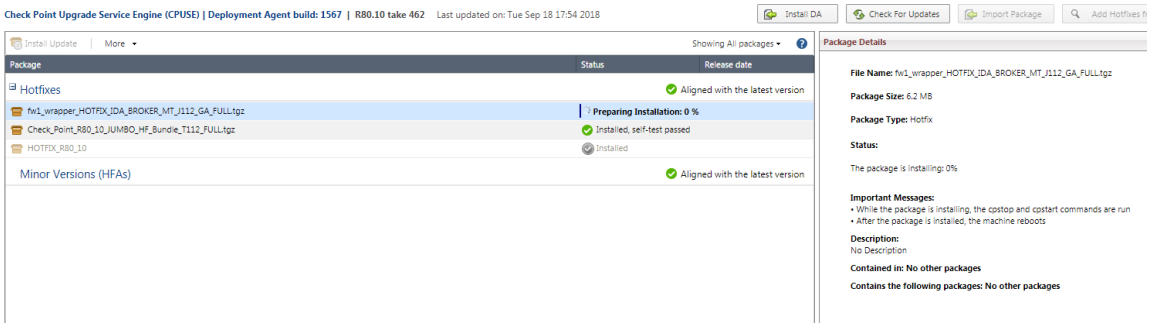
   Note: all settings will be saved once it will be enabled after installation except for portal's accessibility. (Browser based authentication, Identity Agent, Terminal Servers, Identity Collector and Identity Web API). If any of these setting vary from default please reconfigure them after the installation of the PDP Broker HF.
2. Install policy on all Cluster members.
3. Clear all Identity Awareness related tables from all Cluster members at the same time.
   Please run the following command on each Cluster member:
   ```
   fw tab -t pdp_sessions -t pdp_super_sessions -t pdp_encryption_keys -t pdp_whitelist -t pdp_timers -t pdp_expired_timers -t pdp_ip -t pdp_net_reg -t pdp_net_db -t pdp_cluster_stat -t pep_pdp_db -t pep_networks_to_pdp_db -t pep_net_reg -t pep_reported_network_masks_db -t pep_port_range_db -t pep_async_id_calls -t pep_client_db -t pep_identity_index -t pep_revoked_key_clients -t pep_src_mapping_db -t pep_log_completion -x -y
   ```
4. Install the PDP Broker hotfix on each Cluster member using CPUSE (see below for details).
5. Enable Identity Awareness Blade (reconfigure eventually if need the access settings for identity sources as mentioned above) and install access control policy only after hotfix is installed on all Cluster members.

Installing PDP Broker HF using CPUSE

Make sure running the latest version of CPUSE (see sk92449 for details) when you are importing the HF.



Start the installation process and keep in mind a reboot will take place at the end of the process.



The PDP Broker configuration file

Copy a default PDP Broker configuration file to the gateway directory `$FWDIR/conf/identity_broker.C` once the HF is installed. Examples of configuration files can be found in the appendix of this document.

```
[Expert@pdp_broker_1:0]# ls -ltr /opt/CPsuite-R80/fw1/conf/identity_broker.C
-rw-r----- 1 admin root 911 Sep 20 11:48 /opt/CPsuite-R80/fw1/conf/identity_broker.C
[Expert@pdp_broker_1:0]#
```

Verify pdp status

```
[Expert@pdp_broker_1:0]# pdp s s
PDP Daemon status
=================
Daemon start time : 18Sep2018 21:21:23
Policy fetched at : 18Sep2018 21:21:29

Connected PEPs    : 0

[Expert@pdp_broker_1:0]#
```

Install the policy and verify the status again. During the policy installation the configuration file will be read. In case of mistakes in the configurations file, you may need to run debugging as explained in the section “troubleshooting” in this document.

For now keep in mind that relevant logs are written to the pdpd.elg log file.

```
[Expert@pdp_broker_1:0]# cd /opt/CPsuite-R80/fw1/log
[Expert@pdp_broker_1:0]# ls -ltrh pdp*
-rw-rw---- 1 admin root    0 Sep 18 21:35 pdp.elg
-rw-rw-r-- 1 admin root 9.3K Sep 19 08:12 pdpd.elg
```

Verify the status of the PDP Broker instance.

```
[Expert@pdp_broker_1:0]# pdp br s
Remote Subscribers
------------------------------------------------------------------------
| IP | Name | Status | Updates | Filtered | Errors | Last Connectivity |
------------------------------------------------------------------------

Remote Publishers
--------------------------------------------------
| IP | Name | Status | Updates | Last Connectivity |
--------------------------------------------------

[Expert@pdp_broker_1:0]#
```

See a list of all PDP Broker related commands

```
[Expert@pdp_broker_1:0]# pdp broker
Command: root->broker

Available options:
  status          - show status of remote publishers and subscribers
  force_reconnection - try to reconnect to subscriber immediately
  sync_subscriber - sync remote subscriber
  sync_all_subscribers - sync all remote subscribers
  sync_publisher  - sync remote publisher
  sync_all_publishers - sync all remote publishers
  debug           - debug logging menu

[Expert@pdp_broker_1:0]#
```


Preparing the management server or management domain

Modify the `$FWDIR/lib/nac_tables.def` file according to the PDP Broker HF release notes. Before editing the file create a copy.

Change to the `$FWDIR/lib/` and run

```
cp ./nac_tables.def ./nac_tables.def.no_pdp_broker
```

By default the file looks like shown below:

```
// PDP kernel tables
pdp_sessions = dynamic keep sync hashsize 512 limit 50000 kbuf
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25;
pdp_super_sessions = dynamic keep sync hashsize 512 limit 25000 kbuf
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25;
pdp_encryption_keys = dynamic keep sync kbuf 1;
pdp_whitelist = dynamic expires 86400 keep sync hashsize 512 limit 25000;
pdp_timers = dynamic expires 7200 keep sync limit 180000;
pdp_expired_timers = dynamic keep sync;
pdp_ip = dynamic keep sync;
pdp_net_reg = dynamic local_sync keep sync kbuf 1;
pdp_net_db = dynamic local_sync keep sync;
pdp_cluster_stat = dynamic local_sync keep sync expires 30;
pdp_monitor_counters = dynamic keep sync;

//forward declaration for use in NAC-xxx services
pep_configuration = static;
pep_configuration_ips = static;
pep_alias_ips = static;

// END: PDP kernel tables
```

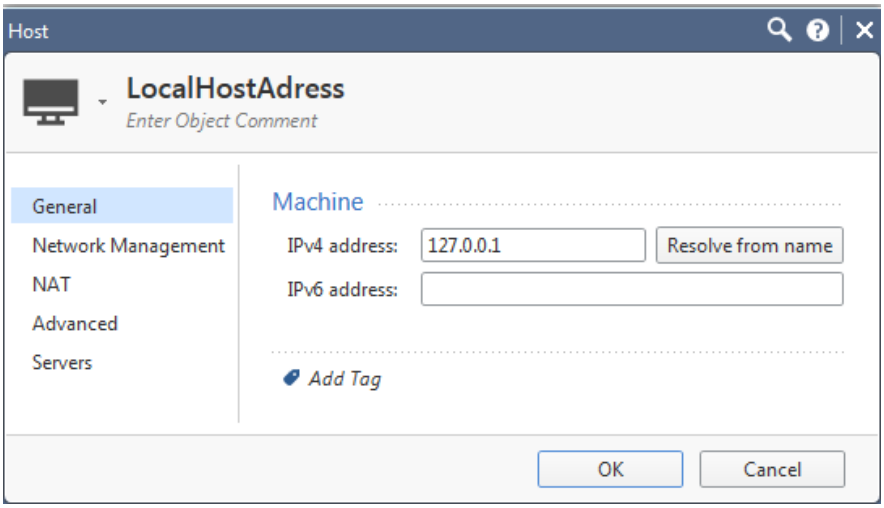
Add the **highlighted** entry

```
// PDP kernel tables
pdp_sessions = dynamic keep sync hashsize 512 limit 50000 kbuf
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25;
pdp_super_sessions = dynamic keep sync hashsize 512 limit 25000 kbuf
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25;
pdp_encryption_keys = dynamic keep sync kbuf 1;
pdp_whitelist = dynamic expires 86400 keep sync hashsize 512 limit 25000;
pdp_timers = dynamic expires 7200 keep sync limit 180000;
pdp_expired_timers = dynamic keep sync;
pdp_ip = dynamic keep sync;
pdp_net_reg = dynamic local_sync keep sync kbuf 1;
pdp_net_db = dynamic local_sync keep sync;
pdp_cluster_stat = dynamic local_sync keep sync expires 30;
pdp_monitor_counters = dynamic keep sync;
pdp_publishers_last_update = dynamic keep sync;

//forward declaration for use in NAC-xxx services
pep_configuration = static;
pep_configuration_ips = static;
pep_alias_ips = static;

// END: PDP kernel tables
```

There is no need restarting the management server as the table file will be read when installing policy on the gateway.

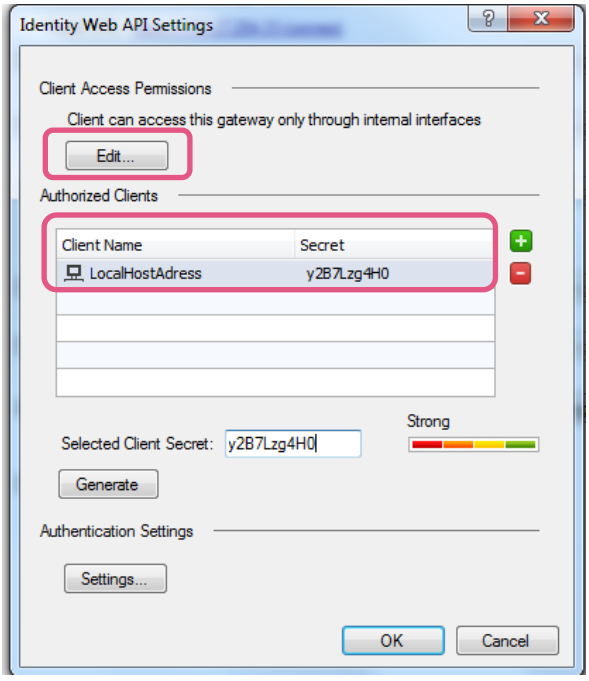
Configuring the gateway object for the required PDP Broker subscriber functionality

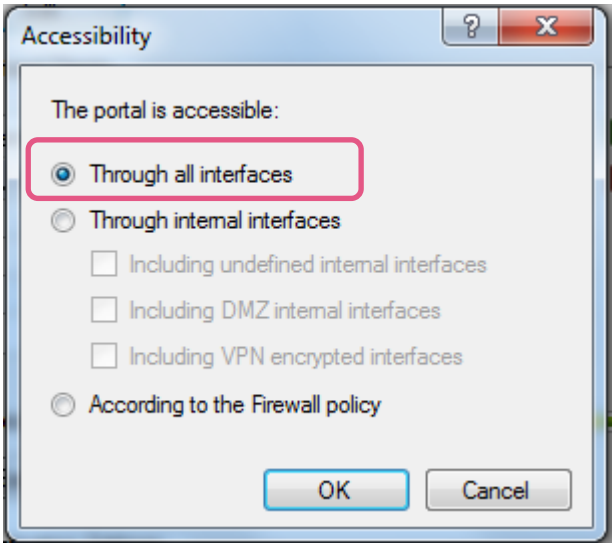
Create a network object describing the local host loopback address.



Enable the ID Awareness Web API and add the local host object to access object to the allowed hosts. This configuration step allows the PDP Broker process accessing the ID Awareness API.

Verify if client access permissions “Accessibility” settings are correct for your network topology.

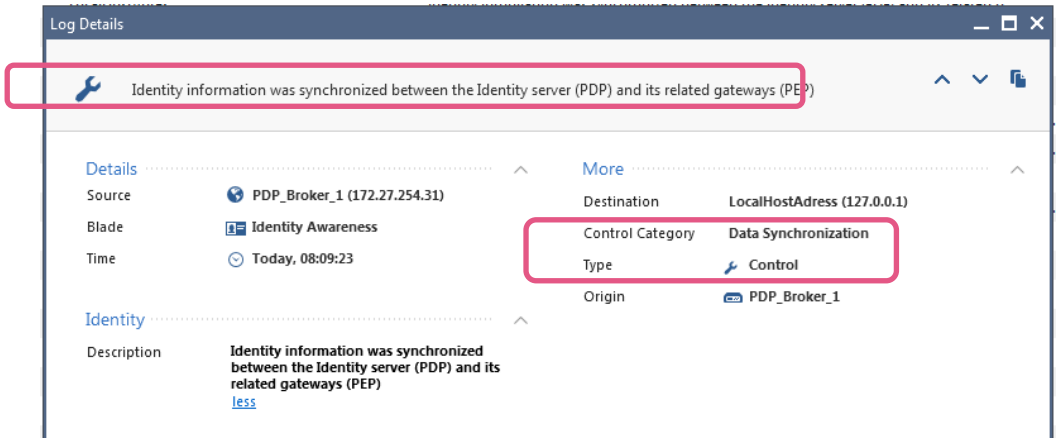




Install the security policy on the PDP Broker gateway.

Verify PDP to PEP identity sharing

Once the policy installation process has been completed, check SmartConsole > Logs & Views for a log message reporting a successful “Data Synchronization”.



Preparing the trust relationship between PDP Brokers

Trust is based on HTTPS connections between the PDP Brokers. The PDP Broker Publisher establishes and HTTPS connection in the direction of the PDP Broker Subscriber. The PDP Broker Subscriber will present its Multi-Portal HTTPS certificate in the TLS handshake. The PDP Broker Publisher must be able to verify the certificate presented. Therefore you need to fetch the Multi-Portal HTTPS certificate and learn the fingerprint of the peering PDP Broker Subscriber(s). In the below example the “pdp_broker_1” fetches the certificate information of “pdp_broker_2” using IP address 172.27.254.32 of the gateway pdp_broker_2.

The command `BrokerCertFetcher` is located in `$FWDIR/bin`.

```
[Expert@pdp_broker_1:0]# /opt/CPsuite-R80/fw1/bin/BrokerCertFetcher 172.27.254.32
Subject:         /O=R8010Mgnt..pobipt/CN=PDP_Broker_2 VPN Certificate
Fingerprint:     BAWL YOGA IVAN DANK FLUE DAR DIN IRON TOM VOID LAG DING
```

Use the “Main IP address” fetching the certificates as this IP address is part of the file name storing the fetched information. In addition the PDP Broker configurations file is containing a reference to the PDP Broker peer using the “Main IP address”.

Important note for clustered environments: make sure using the cluster IP address running the “BrokerCertFetcher” command!

The result (Subject and Fingerprint) are displayed and must be entered in the configuration file at a later stage. For convenience you may want to copy them as well to a scratch board. The file containing the fetched information is stored in `$FWDIR/nac/broker_ca_certs` directory. This directory “`broker_ca_certs`” is created as soon as the `BrokerCertFetcher` has been used the first time.

```
[Expert@pdp_broker_1:0]# ls -ltr /opt/CPsuite-R80/fw1/nac/broker_ca_certs/
-rw-rw---- 1 admin root 1042 Sep 19 13:21 172.27.254.32.pem
```


Working with certificates signed by external CAs or intermediate (sub) CAs

Often customers are using certificates signed by external CAs for the Identity Awareness Captive Portal.

In this case a PKCS#12 container is created by the customers CA and handed over to the Check Point security admin. A PKCS#12 container includes:

- The end entity (PDP Broker) certificate and related private and public keys
  - Make sure this certificate includes a Subject Alternative Name carrying a DNS entry (the FQDN of the PDP Broker)
- The sub-CA/Intermediate CA certificate (and public key) that has signed the end entity certificate
- The root-ca certificate and its related public key

A shortcoming exists in the R80.10 / R80.20 SmartConsole when importing a PKCS#12 container causing an incorrect import of the certificate chain.

As a workaround you need to create a certificate chain file including Root-CA certificate and the Intermediate CA certificate and copy this to the PDP Broker gateway.

Here is what you need to do:

Create a file including the certificate chain of intermediate CA certificate and the root CA certificate in base64 (PEM) format

1) Open the PKCS#12 container on a Windows machine
2) Import the certificate, intermediate CA and root CA certificates
3) Export intermediate and root CA certificates in base64 encoded (PEM) format each to dedicated files
4) Copy both files to the PDP Broker gateway into a temp directory
5) Concatenate both files to one file using the following Linux command

   ```
   cat ./sub-ca.pem ./root-ca.pem > ./certchain.pem
   ```

6) Copy this certificate chain file to the `$FWDIR/nac/broker_ca_certs/` directory assigning the IP address of the PDP Broker peer as file name

   ```
   cp ./certchain.pem $FWDIR/nac/broker_ca_certs/<ip address of PDP Broker peer>.pem
   ```

7) Verify the TLS connectivity running below command (note file `<ip address of PDP Broker peer>.pem` actually contains the certificate chain!)

   ```
   # cpopenssl s_client -connect <IP Addr PDP Broker Peer:443> -CApath /opt/CPsuite-R80/fw1/nac/broker_ca_certs/ -CAfile /opt/CPsuite-R80/fw1/nac/broker_ca_certs/<IP Addr PDP Broker Peer>.pem
   ```

You will see an "OK" code at the end of the output.

Creating the PDP Broker configuration file

Modify an example configuration file listed in the appendix according to your scenario following the guidelines given in the relevant section of this guide and the PDP Broker HF release notes. You may want using a Windows computer running Notepad++ for convenience, copy the file to the `$FWDIR/conf` directory afterwards and running a “dos2unix” conversion. You may as well use the `vi` editor on the gateway itself.

In the following you see a configuration file copied using `scp` to the gateway and then copied to the relevant directory. Once the policy is installed the configuration is active.

```
[Expert@pdp_broker_1:0]# ls -ltr /home/scpuser/identity_broker.C
-rw-r----- 1 scpuser users 1193 Sep 19 11:29 /home/scpuser/identity_broker.C
[Expert@pdp_broker_1:0]# cp /home/scpuser/identity_broker.C /opt/CPsuite-R80/fw1/conf/
[Expert@pdp_broker_1:0]# ls -ltr /opt/CPsuite-R80/fw1/conf/identity_broker.C
-rw-r----- 1 admin root 1193 Sep 19 11:48 /opt/CPsuite-R80/fw1/conf/identity_broker.C
[Expert@pdp_broker_1:0]# dos2unix /opt/CPsuite-R80/fw1/conf/identity_broker.C
dos2unix: converting file /opt/CPsuite-R80/fw1/conf/identity_broker.C to UNIX format
```


Maintain a diagram and list of trust related information

Even in a small environment it is recommended maintaining documentation about the information of each PDP Broker and related trust relationships. See below an example.


```
pdp broker 1                                          pdp broker 2
sharing id          AAbbCCdd000C29F08D40               sharing id          AAbbCCdd000C2967E1BB

Name PDP_Broker_1                                     Name PDP_Broker_2
Main IP Address 172.27.254.31                         Main IP Address 172.27.254.32

Subject:        /O=R8010Mgnt..pobipt/CN=PDP_Broker_1 VPN  Subject:        /O=R8010Mgnt..pobipt/CN=PDP_Broker_2 VPN
Certificate                                           Certificate
Fingerprint:    BAWL YOGA IVAN DANK FLUE DAR DIN IRON     Fingerprint:    BAWL YOGA IVAN DANK FLUE DAR DIN IRON TOM
TOM VOID LAG DING                                     VOID LAG DING

eth0 192.168.169.31/24                                eth0 192.168.169.32/24
eth1 172.27.254.31/24                                 eth1 172.27.254.32/24
```

You may want to add information for filtering later on.

Details about the PDP Broker configurations file

The configuration of the PDP Broker is stored in a text file.

```
[Expert@pdp_broker_1:0]# ls -ltr /opt/CPsuite-R80/fw1/conf/identity_broker.C
-rw-r----- 1 admin root 911 Sep 20 11:48 /opt/CPsuite-R80/fw1/conf/identity_broker.C
```

The file contains

- Information identifying the local PDP Broker instance
    - Sharing_ID: a 20 digit hex value used to identify the PDP Broker
        - You may want to use the MAC address of the management interface combined with an identifier for example : AAbbCCdd000C29F08D40
- Relevant information to establish peering(s) with other PDP Broker(s)
    - Name as defined in SmartConsole
    - Main IP address
    - Subject name as stated in Multi-Portal HTTPS certificate
    - Fingerprint of CA that has signed the Multi-Portal HTTPS certificate
    - IP addresses of interfaces
- Filter information to exclude or include identities published to subscriber(s) from specific sources such as
    - “Identity Collector”, “Identity Agent”, “portal”
    - Active Directory domains described as REGEX example:

```
:include_domains (
    :(“domain1”)
    :(“regexp:mydomain*”)
)
```

Please consult the release notes of the PDP Broker HF for a full list of filter capabilities and see the appendix of this document for examples.

Keep in mind installing the policy in order to have the configuration file being read.

Monitoring and managing PDP Broker using the command line

The PDP Broker can be monitored using a series of CLI commands.

```
[Expert@pdp_broker_1:0]# pdp broker <press enter to see a list of commands>
Command: root->broker

Available options:
  status                - show status of remote publishers and subscribers
  force_reconnection    - try to reconnect to subscriber immediately
  sync_subscriber       - sync remote subscriber
  sync_all_subscribers  - sync all remote subscribers
  sync_publisher        - sync remote publisher
  sync_all_publishers   - sync all remote publishers
  debug                 - debug logging menu
```

Monitoring PDP Broker peering and synchronization of identity updates

Verify the PDP Broker peering on the PDP Broker Publisher

```
[Expert@pdp_broker_1:0]# pdp broker status
Remote Subscribers
-----------------------------------------------------------------------------------------------
| IP            | Name         | Status    | Updates | Filtered | Errors | Last Connectivity  |
-----------------------------------------------------------------------------------------------
| 172.27.254.32 | PDP_Broker_2 | Connected | 4       | 2        | 7      | 20Sep2018 13:40:33 |
-----------------------------------------------------------------------------------------------

Remote Publishers
-----------------------------------------------------
| IP | Name | Status | Updates | Last Connectivity |
-----------------------------------------------------

[Expert@pdp_broker_1:0]#
```

Verify the PDP Broker peering on the PDP Broker Subscriber

```
[Expert@pdp_br_2:0]# pdp br s
Remote Subscribers
-------------------------------------------------------------------------
| IP | Name | Status | Updates | Filtered | Errors | Last Connectivity |
-------------------------------------------------------------------------

Remote Publishers
-------------------------------------------------------------------------------
| IP            | Name         | Status    | Updates | Last Connectivity  |
-------------------------------------------------------------------------------
| 172.27.254.31 | PDP_Broker_1 | Connected | 4       | 20Sep2018 13:41:13 |
-------------------------------------------------------------------------------

[Expert@pdp_br_2:0]#
```

Short version of “pdp broker status”

Troubleshooting

Situation A: PDP Broker doesn´t even try to establish a peering

In case the command “pdp br s” doesn´t show an active peering relationship check if the at least the attempt establishing an outbound https connection is made. Perform a `tcpdump` on the outbound interface in the direction of the PDP Broker peer.

If you don´t see a connection attempt, then most likely parsing the configuration files hasn´t been successful. In this case you best debug the pdpd process while installing the policy.

Enable debugging for pdpd process:

```
[Expert@pdp_broker_1:0]# unset TMOUT
[Expert@pdp_broker_1:0]# echo --- debug start -- >> /opt/CPsuite-R80/fw1/log/pdpd.elg
[Expert@pdp_broker_1:0]# pdp debug set all all
Debug is now turned ON
The following topics were added:
   topic      severity
  ====================
 + all        all

[Expert@pdp_broker_1:0]#
```

<< Install policy >>

```
[Expert@pdp_broker_1:0]# echo --- debug stop -- >> /opt/CPsuite-R80/fw1/log/pdpd.elg
[Expert@pdp_broker_1:0]# pdp debug off
```

Check `pdpd.elg` file for parsing errors by searching for “`read: called with identity_broker.C`”. Often you find a parsing error that needs to get corrected in the configurations file.

Situation B: PDP Broker peering is not established successful

Check the last error reported while trying to establish a PDP Broker peering using the “`-e`” option of “`pdp br s`” command. The below output has been edited to fit the format of this document.

```
[Expert@pdp_broker_1:0]# pdp broker status -e
Remote Subscribers
-----------------<->------------------------------------------------------------>
| IP             |   | Last Error                                                 |
-----------------<->------------------------------------------------------------>
| 172.27.254.32 |   | 20Sep2018 12:06:23 Subscriber has no Broker: Http Error |
-----------------<->------------------------------------------------------------>

->-------------------------------------------------
 Updates | Filtered | Errors | Last Connectivity   |
->-------------------------------------------------
  4      | 2        | 7      | 20Sep2018 13:48:06  |
->-------------------------------------------------

Remote Publishers
-------------------------------------------------
| IP | Name | Status | Updates | Last Connectivity |
-------------------------------------------------
```

In this case the issue may be related to a security gateway blocking the HTTPS connection towards the PDP Broker peer. In addition the PDP Broker Subscriber gateway configuration related to the ID Awareness API might not be configured allowing incoming access requests from all interfaces. Check as well the Sharing Identifiers are correctly reflected in the configuration file.

Situation C: PDP Broker Subscriber isn´t receiving identity updates

In case your PDP Broker Subscriber has a peering to the PDP Broker Publisher but you don´t see Identities being shared (output of "`pdp monitor all`" is different on both peers) you may want to manually initiate the synchronization process between the PDP Broker peers. For this you run the "`force sync`" command on the PDP Broker Publisher.

```
[Expert@pdp_broker_1:0]# pdp br sync_subscriber 172.27.254.32
Sync operation to the remote subscriber has been started
[Expert@pdp_broker_1:0]#
```

Depending on the number of identities to be shared the PDP Broker Subscriber will learn the identities and the output of "`pdp monitor all`" should result being the same like on the PDP Broker Publisher.

Situation D: PDP Broker Subscriber showing an identity just for a short time

Especially when testing in small environments using one computer connecting to both PDP Broker peers you may run into a situation, where a learned identity from one source is getting overwritten by the identity learned by another source having a higher rank.

Example:

PDP Broker 1 and PDP Broker 2 are both working as Publisher and Subscriber and no filtering is applied

When PDP Broker 1 learns an identity "Alice" using the Captive Portal related to source IP address 192.168.169.1 it will share this identity to PDP Broker 2. Running "`pdp m a`" on both PDP Brokers will show you the learned identity "Alice".

If you now connect an ID Agent running on computer 192.168.169.1 to PDP Broker 2 to have PDP Broker learning another identity "Bob" related to the same source IP address this will lead to having PDP Broker 2 dismissing the learned identity "Alice". Then PDP Broker 2 will share this identity update with PDP Broker 1 and even here the identity "Alice" will be removed.

This sequence is the normal mode of operation as both identities are related to the same source IP address. The PDP is rating the source ID Agent higher than the source Captive Portal and as a consequence remove the earlier learned identity.

Appendix

Appendix A: PDP Broker configurations file “publisher only”

```
(
:sharing_id (AAbbCCdd000C29F08D40)
:identity_subscribers (
    : (
            :Name (PDP_Broker_2)
            :sharing_id (AAbbCCdd000C2967E1BB)
            :ipaddr (172.27.254.32)
            :connect_ip ()
            :interfaces (
                    :0 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (192.168.169.32)
                            )
                    :1 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (172.27.254.32)
                            )
            )
            :path_prefix ("/_IA_API/_b")
            :external_port (443)
            :crl_validation_config (fail_open)
            :certificate_subject ("/O=R8010Mgnt..pobipt/CN=PDP_Broker_2 VPN Certificate")
            :certificate_fingerprint ("BAWL YOGA IVAN DANK FLUE DAR DIN IRON TOM VOID LAG DING")
            :outgoing_filter ()
            )
)
:identity_publishers ()
:outgoing_filter (
    :include_users_and_machines ()
    :exclude_users_and_machines ()
                    :include_networks ()
                    :exclude_networks ()
                    :include_identity_source ()
                    :exclude_identity_source ()
                    :include_domains ()
                    :exclude_domains ()
    )
    :incoming_filter ()
)
```

Make sure to have a space between : and (

() indicates “there is no publisher peering”

Appendix B: PDP Broker configurations file “subscriber only”

```
(
:sharing_id (AAbbCCdd000C2967E1BB)
:identity_subscribers ()
:identity_publishers (
    : (
            :Name (PDP_Broker_1)
            :sharing_id (AAbbCCdd000C29F08D40)
            :ipaddr (172.27.254.31)
            :recalculate_access_roles (false)
            :interfaces (
                    :0 (
                            :netmask (255.255.255.0)
                            :ipaddr (192.168.169.31)
                            )
                    :1 (
                            :netmask (255.255.255.0)
                            :ipaddr (172.27.254.31)
                            )
            )
            :incoming_filter ()
            )
    )
    :outgoing_filter (
            :include_users_and_machines ()
            :exclude_users_and_machines ()
            :include_networks ()
            :exclude_networks ()
            :include_identity_source ()
            :exclude_identity_source ()
            :include_domains ()
            :exclude_domains ()
            )
    :incoming_filter ()
)
```

() indicates “there is no subscriber peering”

Appendix C: PDP Broker 1 configured as publisher and as subscriber

```
(
:sharing_id (AAbbCCdd000C29F08D40)
:identity_subscribers (
    : (
            :Name (PDP_Broker_2)
            :sharing_id (AAbbCCdd000C2967E1BB)
            :ipaddr (172.27.254.32)
            :connect_ip ()
            :interfaces (
                    :0 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (192.168.169.32)
                            )
                    :1 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (172.27.254.32)
                            )
            )
            :path_prefix ("/_IA_API/_b")
            :external_port (443)
            :crl_validation_config (fail_open)
            :certificate_subject ("/O=R8010Mgnt..pobipt/CN=PDP_Broker_2 VPN Certificate")
            :certificate_fingerprint ("BAWL YOGA IVAN DANK FLUE DAR DIN IRON TOM VOID LAG DING")
            :outgoing_filter ()
            )
)
:identity_publishers (
    : (
            :Name (PDP_Broker_2)
            :sharing_id (AAbbCCdd000C2967E1BB)
            :ipaddr (172.27.254.32)
            :recalculate_access_roles (false)
            :incoming_filter (
                                    :exclude_domains ()
                                    :exclude_identity_source ()
                                    :exclude_users_and_machines ()
                                    :exclude_networks ()
                                    :include_networks ()
                                    :include_identity_source ()
                                    :include_users_and_machines ()
                                    )
            :interfaces (
                    :0 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (192.168.169.32)
                            )
                    :1 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (172.27.254.32)
                            )
        )
            )
)
:outgoing_filter (
    :include_users_and_machines ()
    :exclude_users_and_machines ()
                    :include_networks ()
                    :exclude_networks ()
                    :include_identity_source ()
                    :exclude_identity_source ()
                    :include_domains ()
                    :exclude_domains ()
    )
    :incoming_filter ()
)
```


Appendix D: PDP Broker 2 configured as publisher and as subscriber

```
(
:sharing_id (AAbbCCdd000C2967E1BB)
:identity_subscribers (
    : (
            :Name (PDP_Broker_1)
            :sharing_id (AAbbCCdd000C29F08D40)
            :ipaddr (172.27.254.31)
            :connect_ip ()
            :interfaces (
                    :0 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (192.168.169.31)
                            )
                    :1 (
                                    :netmask (255.255.255.0)
                                    :ipaddr (172.27.254.31)
                            )
            )
            :path_prefix ("/_IA_API/_b")
            :external_port (443)
            :crl_validation_config (fail_open)
            :certificate_subject ("/O=R8010Mgnt..pobipt/CN=PDP_Broker_1 VPN Certificate")
            :certificate_fingerprint ("BAWL YOGA IVAN DANK FLUE DAR DIN IRON TOM VOID LAG DING")
    )
)
:identity_publishers (
    : (
            :Name (PDP_Broker_1)
            :sharing_id (AAbbCCdd000C29F08D40)
            :ipaddr (172.27.254.31)
            :recalculate_access_roles (false)
            :incoming_filter (
                                    :exclude_domains ()
                                    :exclude_identity_source ()
                                    :exclude_users_and_machines ()
                                    :exclude_networks ()
                                    :include_networks ()
                                    :include_identity_source ()
                                    :include_users_and_machines ()
            )
            :interfaces (
                    :0 (
                            :netmask (255.255.255.0)
                            :ipaddr (192.168.169.31)
                            )
                    :1 (
                            :netmask (255.255.255.0)
                            :ipaddr (172.27.254.31)
                            )
            )
    )
)
:outgoing_filter (
    :include_users_and_machines ()
    :exclude_users_and_machines ()
    :include_networks ()
                    :exclude_networks ()
                    :include_identity_source ()
                    :exclude_identity_source ()
                    :include_domains ()
                    :exclude_domains ()
    )
    :incoming_filter ()
)
```

Appendix E: PDP Broker software library used in tests documented here

```
[Expert@pdp_broker_1:0]# cpvinfo /opt/CPsuite-R80/fw1/lib/libpdplib.so

** Version info attributes of '/opt/CPsuite-R80/fw1/lib/libpdplib.so' **

Type = library
Name = pdplib
Module Name = NACServer
Build Number = 991003006
Major Release = NGX
Minor Release = ida_broker_mt_j112
Release Number = 5.0.5
Version Name = NGX
Interface Version = 0
Implementation Version = 6
Internal Name = pdplib
Configuration = linux50/release.dynamic
Comments = NULL
Company Name = Check Point Software Technologies LTD.
Legal Copyright = (c) 2005-2009 Copyright Check Point Software Technologies Ltd
[Expert@pdp_broker_1:0]#
```

