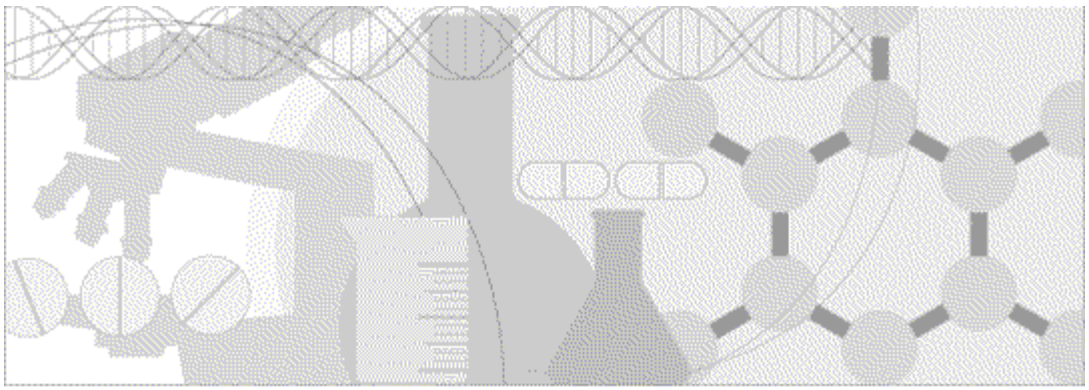
Secure Development Guide

Oracle® Health Sciences InForm 6.1.1



ORACLE

Part number: E72493-01

Contents

CHAPTER 1

Security overview

In this chapter

OWASP top ten security vulnerabilities 2013

To guide developers for what they need to protect against, the Open Web Application Security Project publishes an annual document that lists the ten most critical security vulnerabilities identified for a year. Addressing the ten security vulnerabilities does not provide for total security, but is a good start in raising awareness to the current major security threats. This document explains how the Clinical Data API and API developers should address security vulnerabilities and risks documented by OWASP for 2013.

This document identifies the controls within the Clinical Data API that are used or may be used to address the associated risks. In some cases, the controls are baked into the product and proper use of the controls by the clients must be used to validate the integrity of the controls.

Security awareness and education

The best application security money can buy is education. Developers and project leads need to be mindful of security issues and have an understanding of secure coding practices. Training must include an in depth explanation of the potential risks as well as features of the development and deployment platforms that help mitigate exploits.

The most important design principle for application security is to implement security by design and default. Secure coding guidelines should be made available, adhered to, and enforced in all development organizations, irrespective of the tools and platforms being used.

A good example for security by default is the expectation that we all have for how elevators behave in case of a power outage. Instead of releasing the breaks, we expect elevators to apply the breaks for the safety of passengers in the cabin. But how would the elevator know that it should apply the brakes if no one defined this as the default behavior? So before thinking about how to prevent external attacks, it makes sense to identify secure defaults for an application to protect it from the inside. This however does not work well without training and awareness.

The risk associated with build your own

It is not always that developers immediately find the security they need for an application within the security toolset provided by a platform or built into a framework. As a result, *build your own security* is not uncommon among development projects. This is especially true if the application is a replacement of an existing system that uses a specific non-standard security infrastructure. An example for this is database-table-based authentication and authorization in combination with user provisioning and resource granting at runtime.

The risk associated with building your own security is that you are also on your own when it comes to quality assurance of the security layer, application security propagation, and single sign on, and you are responsible for bug fixing and maintenance of the security layer.

Not all developers are security experts, but experts are what it takes to build a custom security layer.

Time spent investigating existing, well-vetted security solutions is probably time well spent. Existing solutions can be applied to custom applications more easily and more cost effectively than creating an error-prone, self-written mechanism.

Other aspects of security

Application security is useless if the application itself runs in an insecure environment. Perimeter security describes the levels of protection that are added on servers, the network, and other data access channels outside of the API domain. As can be seen in this document, not all of the OWASP top ten security vulnerabilities documented for 2013 are relevant for application developers for specific implementations.

CHAPTER 2

Top ten security risks for 2013

In this chapter

Overview of the OWASP top ten list

The OWASP top ten list for the year 2013 does not differ much from lists published for previous years, except for changes in ranking. The listed security threats are probably the most severe threats and application developers have to be aware of and protect against these threats.

For more information, see the following:

- OWASP home page:

  https://www.owasp.org/index.php/Main_Page

- General descriptions for the OWASP top ten list of security risks for 2013:

  https://www.owasp.org/index.php/Top_10_2013-Top_10

- Overview of the security risks at:

  https://www.owasp.org/index.php/Top_10_2013-Risk

#1 - Injection

Injection vulnerabilities occur when data is sent into an interpreter via an interface specification and the party submitting the data does not perform checks on the data to ensure only the expected actions are performed by the interpreter on the data. SQL, Code, Command, Log, Path Transversal (XML) are all possible types of injection based upon the interpreter used in the container.

Valid content types

The Clinical Data API is a web service based on SOAP over HTTP. Developers must use *application/soap+xml* as MIME types in the Content-Type HTTP headers. Otherwise, the API rejects the requests.

SQL injection

To prevent SQL injections, the Clinical Data API uses bind variables and does not dynamically generate SQL, which makes SQL injection impossible.

XML injection

The Clinical Data API handles XML injections by using standard XML processing components that construct the XML documents. Oracle recommends that client code also uses standard XML processing components to ensure that data is properly encoded. If XML is constructed manually, the developer needs to ensure that any untrusted data is properly encoded to prevent XML injection. As a best practice, the developers must validate the XML against the XML schema provided by the Clinical Data API, as the Clinical Data API does to ensure that the constraints for the data type and length are met.

#2 - Broken authentication and session management

Risks associated with broken authentication and session management are often due to these functions not being implemented properly. As previously stated, custom authentication mechanisms should not be implemented and have not been implemented. To address web service client authentication attacks, the Clinical Data API supports username token authentication. To ensure the integrity of web client authentication, the proper handling of the authentication artifacts should be followed.

To ensure the web client authentication is secure, the password for the username token should be treated with the utmost care since exposure of the password could compromise the authentication mechanisms systems. The Clinical Data API does not store the password in clear-text on the file system and does not log the password. As such, the client password should be protected in the same way. The password should always be stored in an encrypted fashion. Do not transfer the password through un-encrypted side channels between web service endpoint parties when exchanging the password to reduce exposure. The authentication of each side channel endpoint is also a concern during the exchanging of the password and is open to social engineering attacks if not done properly. To access the web service interfaces and to use the Clinical Data API, you must be an InForm Integration user with the ODM Submit right. For more information, see the *Clinical Data API Guide*.

The Clinical Data API is stateless and does not maintain the session. The API is re-entrant and the same credentials may be used for the calls. Considerations with the number of the concurrent calls should be designed not to exhaust the resources of the systems.

#3 - Cross site scripting (XXS)

Cross site scripting occurs due to browser presentation of data. Although web services generally do not support web containers, all input data to the API must be properly validated or escaped. Generally this vulnerability is not applicable to web services.

#4 - Insecure direction object references

When a developer exposes a reference to an object without proper access or other protection, then this reference becomes a source of attack. The objects defined in the Clinical Data API have been tested to validate proper authorization constructs within the functions of the defined service. When developing code and sending data to and from the API, ensure that the authorization model of the API interface is consistent to guard against insecure direction object references.

#5 - Security misconfiguration

To securely allow subject data to be submitted, the Clinical Data API requires authentication information from two users:

- An InForm Integration user with the ODM Submit right.
- An InForm Site or Sponsor user with the Enter CRF data right.

#6 - Sensitive data exposure

Not all data is public and caution should be used to hide sensitive information from unauthorized users. Failure in security configuration and the selection of insecure defaults may pose a of risk data leakage.

Developers should use TLS 1.2 or above to consume the Clinical Data API to ensure the protection of the sensitive data and address Man-in-the-Middle attacks. Web client developers should enforce encrypted data transport when the application transports sensitive data and should validate that all certificates are legitimate and signed by public authorities. Ciphers should be restricted to modern implementations.

#7 - Missing function level access control

The defense in depth design pattern specifies that multiple layers of security must be implemented in an application. This also means that application functionality that executes methods and operations should be guarded by authorization checks even if the underlying data object is protected through entity security. When the client application calls out to the Clinical Data API to submit the data, such calls should be protected with the level of the access control. As a best practice, never assume that a specific method will only be called within the context that it was initially designed for. All access to functionality that manipulates data must be protected either by access control on the entity or by guarding the invocation of methods with the appropriate permission checks. The credential of the identity associated with the access control in the client application must be encrypted and stored in the secured identity management system as the API does.

#8 - Cross-site request forgery (CSRF)

Cross-site request forgery requires a browser container. Generally APIs are not meant to be supported directly in a browser container so the session is not kept as a browser cookie and CSRF is not a viable threat.

#9 - Using components with known vulnerabilities

The Clinical Data API stack is constantly updated with the latest security fixes and patches. Oracle recommends that developers using the API do the same.

#10 - Non-validated redirects and forwards

Clinical Data API responses do not provide URLs to other resources or sub-resources in the representations that are returned. Therefore, the API is not subject to this vulnerability.

About the documentation

Where to find the product documentation

The product documentation is available from the following locations:

- **My Oracle Support** (https://support.oracle.com)—*Release Notes* and *Known Issues*.
- **Oracle Technology Network** (http://www.oracle.com/technetwork/documentation/hsgbu-154445.html)—The most current documentation set, excluding the *Release Notes* and *Known Issues*.

If the software is available for download, the complete documentation set is available from the Oracle Software Delivery Cloud (https://edelivery.oracle.com).

All documents may not be updated for every InForm release. Therefore, the version numbers for the documents in a release may differ.

Documentation accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Documentation

Document	Description	Part number	Last updated
Release Notes	The *Release Notes* document describes enhancements and issues fixed in the current release, and other late-breaking information.	E61316-01	6.1.1
Known Issues	The *Known Issues* document provides detailed information about the known issues in this release, along with workarounds, if available.	E61318-01	6.1.1

Document	Description	Part number	Last updated
Secure Configuration Guide	The *Secure Configuration Guide* provides an overview of the security features provided with the Oracle® Health Sciences InForm application, including details about the general principles of application security, and how to install, configure, and use the InForm application securely.	E61320-01	6.1.1
Upgrade and Migration Guide	The *Upgrade and Migration Guide* provides instructions for upgrading and migrating the InForm software and InForm Portal software to the current InForm release, and for upgrading the Cognos software for use with the Reporting and Analysis module.	E61321-01	6.1.1
Installation Guide	The *Installation Guide* describes how to install the software and configure the environment for the InForm application and Cognos software.	E61322-01	6.1.1
Study and Reporting Setup Guide	The *Study and Reporting Setup Guide* describes how to perform the tasks that are required to set up an InForm study and configure the Reporting and Analysis module for the study.	E61323-01	6.1.1
User Guide	The *User Guide* provides an overview of the InForm application including details on multilingual studies, how to navigate through the user interface, how to manage a study-specific Home page with the InForm Portal application, and how to accomplish typical tasks you perform while running a clinical study. This document is also available from the user interface.	E61324-01	6.1.1
Reporting and Analysis Guide	The *Reporting and Analysis Guide* provides an overview of the Reporting and Analysis module. It includes a brief overview of the Reporting and Analysis interface, illustrates how to access the InForm Ad Hoc Reporting workspace, and describes the study management and clinical data packages available for creating reports. It also provides detailed descriptions of each standard report that is included with your installation.	E61326-01	6.1.1
Reporting Database Schema Guide	The *Reporting Database Schema Guide* describes the Reporting and Analysis database schema, and provides information on creating Reporting Database Extracts (RDEs).	E61327-01	6.1.1

Document	Description	Part number	Last updated
InForm Utilities Guide	The *InForm Utilities Guide* provides information about and step-by-step instructions for using the following utilities: • PFConsole utility • MedML Installer utility • InForm Data Import utility • InForm Data Export utility • InForm Performance Monitor utility • InForm Report Folder Maintenance utility This guide also provides reference information for the MedML elements and scripting objects that are used to import and export data to and from the InForm application, as well as sample data import XML.	E61328-01	6.1.1
MedML Installer utility online Help	The MedML Installer utility online Help provides information about, and step-by-step instructions for using, the MedML Installer utility, which is used to load XML that defines study components into the InForm database. This guide also provides reference information for the MedML elements and scripting objects that are used to import and export data to and from the InForm application, as well as sample data import XML. This document is also available from the user interface.	NA	NA
InForm Data Export utility online Help	The InForm Data Export utility online Help provides information about and step-by-step instructions for using the InForm Data Export utility, which is used to export data from the InForm application to the following output formats: • Customer-defined database (CDD) • Name value pairs This document is also available from the user interface.	NA	NA

Document	Description	Part number	Last updated
InForm Data Import utility online Help	The InForm Data Import utility online Help provides information about and step-by-step instructions for using the InForm Data Import utility, which is used to import data into the InForm application. This document is also available from the user interface.	NA	NA
Clinical Data API Guide	The *Clinical Data API Guide* provides information about submitting data to the InForm application in InForm ODM format.	E61329-01	6.1.1
Third Party Licenses and Notices	The *Third Party Licenses and Notices* document includes third party technology that may be included in or distributed with this product.	E61330-01	6.1.1
Secure Development Guide	The *Secure Development Guide* provides an overview of common security risks for developers using Application Programming Interfaces (APIs) with the Oracle® Health Sciences InForm application, and information on how to address those risks.	E72493-01	6.1.1