Oracle® Health Sciences Clinical One Platform

Security Guide

Release 21.4
F51740-01
January 2022

Oracle Health Sciences Clinical One Platform Security Guide, Release 21.4

F51740-01

Contents

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through Support Cloud.

Contact our Oracle Customer Support Services team by logging requests in one of the following locations:

- English interface of Oracle Health Sciences Customer Support Portal (https://hsgbu.custhelp.com/)
- Japanese interface of Oracle Health Sciences Customer Support Portal (https://hsgbu-jp.custhelp.com/)

You can also call our 24x7 help desk. For information, visit http://www.oracle.com/us/support/contact/health-sciences-cloud-support/index.html or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Preface

This preface contains the following sections:

Additional copyright information

This documentation may include references to materials, offerings, or products that were previously offered by Phase Forward Inc. Certain materials, offerings, services, or products may no longer be offered or provided. Oracle and its affiliates cannot be held responsible for any such references should they appear in the text provided.

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.

Related resources

All documentation and other supporting materials are available on the Oracle Help Center.

1

Security overview

OWASP top ten security vulnerabilities for 2017

To guide developers on what they need to protect against, the Open Web Application Security Project (OWASP) publishes a document that lists the ten most critical security vulnerabilities identified for the year. Addressing these ten security vulnerabilities does not provide for total security, but is a good start in raising awareness on the current major security threats.

This document explains how the Oracle Clinical One Platform API addresses security vulnerabilities, identifies the controls within the Oracle Clinical One Platform API that are used or may be used to address the associated risks, and describes how API developers should address security vulnerabilities and risks documented by OWASP for 2017.

In some cases, the controls are coded into the product and proper use of the controls by API developers is necessary to validate the integrity of the controls. Developers using the Oracle Clinical One Platform API should use this document along with the Oracle Cloud SaaS Security Practices (download link available for registered users in the Oracle Cloud Operations Center on My Oracle Support, Doc ID 1541346.2) to identify the necessary security controls to maintain the risk posture of the API interface while integrating with Oracle Clinical One Platform.

Security awareness and education

The best application security that money can buy is education. Developers and project leads must be mindful of potential 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 of security by default is the expectation that we all have for how elevators behave in case of a power outage. Instead of releasing the brakes, we expect elevators to apply the brakes 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

Developers don't always 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 2017 are relevant for application developers for specific implementations.

2

Top ten security risks for 2017

Overview of the OWASP top ten list

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 2017: https://www.owasp.org/index.php/Top_10-2017_Top_10
- Overview of the security risks: https://www.owasp.org/index.php/Top_10-2017_Application_Security_Risks

#1 - Injection

Injection vulnerabilities can occur when data is sent to an interpreter via an interface specification and the party submitting the data does not check the data to ensure that the interpreter performs only the expected actions 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
- SQL injection
- XML injection

Valid content types

The Oracle Clinical One Platform suite is a microservice-based application built on JAX-RS libraries and hosted in Oracle Health Sciences Cloud (also known as GBUCS).

Developers must use @Produces to handle different Content-Type. Also consider placing @Valid annotation for handling parameters during invocation itself.

SQL injection

The Oracle Clinical One Platform API passes different parameters as bind variables into the named queries, which makes SQL injection impossible.

XML injection

The Oracle Clinical One Platform Data API handles XML injections by using an XML parser library built within the core framework. During integration with other external systems, we recommend that you extend this XML parser library and verify if the definition or constraint for data type and length are accurately met.

#2 - Broken authentication

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. Since the Oracle Clinical One Platform API works in a stateless manner, all inbound requests must either carry a valid JWT token, or perform basic authentication.

The JWT token is encrypted with the necessary algorithm, in accordance with security standards, and it carries the necessary information (expiry, user, principals) in its payload.

When using basic authentication, always transmit credentials over a secure channel.

Credentials are not stored or logged in the Oracle Clinical One Platform system at any point in time.

The Oracle Clinical One Platform API also performs fine-grained authorization for the user to ensure that only a user with the appropriate roles is allowed to access the REST web service endpoints.

#3 - Sensitive data exposure

Not all data is public and we advise that you hide any sensitive information from unauthorized users. Failure in security configuration and the selection of insecure defaults may pose a risk of data leakage.

Developers should use TLS 1.2 or above to consume the Oracle Clinical One Platform API to protect 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.

#4 - XML external entities (XXE)

The Oracle Clinical One Platform API does not use XML for any kind of interaction. However, there is a JAXB implementation available within the core library where XML external entity support has been programmatically turned off as a precaution. We recommend that you implement a similar approach based on recommendations from the OWASP XXE Prevention Cheat Sheet.

#5 - Broken access control

All Oracle Clinical One Platform API endpoints enforce fine-grained authorization for the user, to ensure that only a user with the appropriate roles is allowed to access the REST web service endpoints.

#6 - Security misconfiguration

The Oracle Clinical One Platform suite is always installed in a secure-by-default state in the production environment. Default server users are disabled, and unused ports are blocked as part of deployment. Any communication to Oracle Clinical One Platform application servers happens through the Oracle Health IAMS, and the OHSIAMS WebGate monitors each API call for authentication details.

As part of standard protocol, black box testing is regularly carried out to detect any possible misconfiguration issues. Possible issues are addressed in either the source code or the infrastructure by applying critical patches.

#7 - Cross-site scripting (XSS)

Cross-site scripting occurs chiefly due to browser presentation of data. Input data to the Oracle Clinical One Platform API is always passed through an initialization request filter located within its core library. This filter looks for suspicious or harmful characters, and blocks requests if needed. All outbound responses are decorated with an X_FRAME_OPTIONS header to prevent clickjacking.

#8 - Insecure deserialization

The Oracle Clinical One Platform API validates all inbound requests and serializable objects at multiple levels. No data persists at the middle-tier level. In addition, Oracle Clinical One Platform is hosted within the GBUCS cloud, further reducing the chances of a remote execution scenario. Therefore, insecure deserialization scenarios are not applicable for users of the Oracle Clinical One Platform API.

#9 - Using components with known vulnerabilities

The Oracle Clinical One Platform technology stack is constantly updated with the latest security fixes and patches. We recommend that developers using the API maintain their technology in the same way.

#10 - Insufficient logging and monitoring

All Oracle Clinical One Platform API calls pass through authorization service filters. These filters log any kind of unusual or unexpected behavior, which is shown in the centralized Kibana dashboard. Active monitoring of unexpected events is very important to identify and notify of active attacks so that preventive measures can be taken to address advanced persistent threats (APTs).

3 Summary

In summary, we've discussed the top ten vulnerabilities across the industry and, where applicable, described the security controls that are available to address these vulnerabilities in Oracle Clinical One Platform. Extending these controls into your interface and systems will maintain the security posture of the whole system. While addressing these ten security vulnerabilities does not provide for total security, it is a good start in raising awareness on the current major security threats.

We cannot cover all security vulnerabilities due to the large number of possibilities, but this document provides a list of some of the security controls that should be incorporated into your code to start with and should serve as a catalyst to initiate further investigation into addressing other possible vulnerabilities.