Oracle Life Sciences Identity and Access Management Service

Secure Development Guide

Release 23.2
F84936-01
August 2023

Oracle Life Sciences Identity and Access Management Service Secure Development Guide, Release 23.2

F84936-01

Contents

Preface

1 About this guide

Preface

This preface contains the following sections:

- Additional copyright information
- Documentation Accessibility
- Access to Oracle Support

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.

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.

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 Customer Support Portal (https://hsgbu.custhelp.com/)
- Japanese interface Customer Support Portal (https://hsgbu-jp.custhelp.com/)

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

1

About this guide

This *Secure Development Guide* provides assistance in mitigating common security risks for developers using the Oracle Life Sciences IAMS Inbound User Provisioning Service API, developed based on the SCIM standard and REST framework.

This guide describes how to prevent the main security risks, as identified by the Open Web Application Security Project (OWASP) in their top 10 critical web application security vulnerabilities for 2017, and provides insights for software developers into how the API was created and can be used while addressing these vulnerabilities.

Since in-depth defense is an important strategy for a secure product, do not exclusively rely on the techniques documented in this guide. Implement and extend these techniques in your own code as you develop your interface to the API specification.

> **Note:**
>
> The recommendations in this guide are not exhaustive and no guarantee is offered that implementing all the suggestions provides sufficient protection against all security threats. You cannot delegate responsibility for secure application development to a third party or a single document.
> The purpose of this document is to support developers in knowing the security tools and features that they can use to implement application security when using Oracle Life Sciences IAMS APIs. This document does not replace a formal review process.

About the OWASP Top 10 Security Vulnerabilities for 2017

The Open Web Application Security Project (OWASP) publishes an annual list of the 10 most critical security vulnerabilities identified for the current year to educate developers on the security risks they most likely need to protect against. The OWASP top 10 vulnerability listing is technology agnostic and does not contain language or framework specific examples, explanations, hints, or tips.

This section discusses the practices and strategies used by Oracle Life Sciences IAMS API to mitigate risks posed by the security vulnerabilities documented in the OWASP Top 10 – 2017. Customers using Oracle Life Sciences IAMS APIs should be aware of and protect against these threats. The listed security threats are probably the most severe threats and application developers have to be aware of and protect against these threats.

Addressing these ten security vulnerabilities doesn't provide for total security, but it is a good starting point in preventing the current major security threats. This document explains how the Oracle Life Sciences IAMS Inbound User Provisioning Service API addresses these potential security risks and how API developers should address these security vulnerabilities and risks when using the API.

General descriptions of the top 10 security risks identified by OWASP for 2017 are available at: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_Top_10.html.

You can get an overview of the security risk for an application at: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_Application_Security_Risks.html

About Security Awareness and Education

The best way to ensure application security is through education. Developers and project leaders should be aware of security issues and secure coding practices. Training for these roles should include an in-depth explanation of the potential risks, as well as cover the 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 by default. Secure coding guidelines should be made available, adhered to, and enforced in all development organizations, irrespective of the tools and platforms being used.

An example of security by default is the behavior of elevators 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. The elevator applies the brakes because this was defined as the default behavior.

So, before thinking about how to prevent external attacks, identify secure defaults for an application that can protect it from the inside. This, however, does not work well without training and awareness.

About the Risk Associated with "Build Your Own Security"

Developers don't always immediately identify the security measures 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 its own non-standard security infrastructure. An example for this is database based authentication and authorization in combination with user provisioning and granting access to resources at runtime.

The risk associated with building your own security is that you are also responsible for quality assurance of the security layer, application security propagation and single sign-on, as well as bug fixing and maintenance of the security layer. Not all developers are security experts, but experts are a necessity to build a custom security layer.

We recommend allocating time to investigate and implement existing, well vetted security solutions. Applying existing solutions to custom applications may be easier and more cost-effective than creating custom mechanisms that may offer less protection in their incipient phases.

Addressing the Top Security Risks for Oracle Life Sciences IAMS APIs

The below sections identify the controls within the Oracle Life Sciences IAMS Inbound User Provisioning Service API that are used or may be used to address the top 10 security risks identified by OWASP for 2017.

In some cases, the controls are part of the product and proper use of the controls by the clients is required to validate the integrity of the controls.

The following risks are considered:

- Injection
- Broken Authentication
- Sensitive Data Exposure
- XML External Entities (XXE)
- Broken Access Control
- Security Misconfiguration
- Cross-site Scripting (XSS)
- Insecure Deserialization
- Using Components with Known Vulnerabilities
- Insufficient Logging and Monitoring

Injection

Injection vulnerabilities occur when data is sent into an interpreter via an interface specification and the party submitting the data does not check the data to ensure that only the expected actions are performed on the data by the interpreter.

Injections of the type SQL, code, command, log, path transversal (XML) are all possible, based on the interpreter used in the container.

- Valid Content Types
- SQL Injection
- XML Injection
- LDAP Injection

Valid Content Types

The REST service of the Oracle Life Sciences IAMS Inbound User Provisioning Service supports only the JSON data format. The service client must accept and send 'application/json' as the media type when invoking the REST service.

SQL Injection

To prevent SQL injections, the Oracle Life Sciences IAMS Inbound User Provisioning Service API uses bind variables in SQL queries.

XML Injection

XML injections are not possible because the Oracle Life Sciences IAMS Inbound User Provisioning Service API accepts and generates only JSON data format.

LDAP Injection

The Oracle Life Sciences IAMS Inbound User Provisioning Service API checks all incoming user input through a preset whitelist of allowed input and checks the pattern of the ultimate payload for any attack. Although we generally do not recommend making use of custom code, you can further enhance protection against LDAP injections through custom code on the client side.

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. They have not been implemented for the Oracle Life Sciences IAMS Inbound User Provisioning Service API, which uses a BASIC authentication mechanism. The session is created on request and destroyed at the end of the response. Each API request must be accompanied by BASIC authentication headers to prevent session hijacking.

Sensitive Data Exposure

We recommend hiding sensitive information from unauthorized users and handling sensitive data securely. Failure in security configuration and selecting insecure default settings may facilitate data leakage.

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. Also, ciphers should be restricted to modern implementations.

XML External Entities (XXE)

Oracle Life Sciences IAMS Inbound User Provisioning Service API accepts and generates only JSON data format. It does not accept XML files or use XML processors.

Broken Access Control

When a developer exposes a reference to an object without proper access or other protection, this reference can become a means of attack. 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.

As a best practice, do not assume that a method will only be called within the context for which it was initially designed. 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 at the client application must be encrypted and stored securely.

The authorization model in the SCIM interface ensures protection down to the object level and the SCIM interface has been validated for proper authorization constructs within the functions of the defined service. Any client code built to interact with the SCIM interface should complement this security model so that proper authorization is controlled at the object level.

As a best practice, do not assume that a method will only be called within the context for which it was initially designed. 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 at the client application must be encrypted and stored securely.

Security Misconfiguration

Since Oracle Life Sciences IAMS is hosted in the Oracle Cloud for Industry (OCI) environment, its services adhere to the OCI policies, as described here. Clients of the Oracle Life Sciences IAMS API should implement similar security and available polices to ensure the confidentiality, integrity, and availability of data flowing through the interface.

Cross-site Scripting (XSS)

The Oracle Life Sciences IAMS Inbound User Provisioning Service API prevents cross-site scripting by following the OWASP recommended methodologies.

The best method to address XSS is to validate all data using whitelists and encode data as necessary, according to the presentation or handling of the data.

Insecure Deserialization

Applications and APIs are vulnerable if they deserialize hostile or tampered objects supplied by an attacker. Oracle Life Sciences IAMS Inbound User Provisioning does not accept serialized data from external client applications.

Using Components with Known Vulnerabilities

The technology stack for the Oracle Life Sciences IAMS Inbound User Provisioning Service API is constantly updated with the latest security fixes and patches. Oracle recommends that developers using the API do the same on their end.

Insufficient Logging and Monitoring

Insufficient logging, detection, and monitoring coupled with missing or ineffective integration with incident response, allows attackers to further attack business systems.

Oracle Life Sciences IAMS Inbound User Provisioning Service API access is logged as part of an OIM audit framework. The audit framework provides several audit reports with key details, such as timestamp, source of creation, user ID to identify the source of operation.

Other Aspects of Security

Application security is ineffective 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 and should also be considered to ensure thorough prevention of security risks.

As can be seen in this document, not all of the OWASP top 10 security vulnerabilities for 2017 are relevant for application developers, depending on the implementation.

Recommended Reading

We recommend that you familiarize yourself with the content available on the OWASP website: https://www.owasp.org/index.php/Main_Page

Related Documents

For more information, see the *Oracle Life Sciences Identity and Access Management Service Inbound User Provisioning API Guide* on the Oracle Help Center.

> **Note:**
>
> Always check the Oracle Help Center to ensure you have the latest documentation.