Oracle® Healthcare Data Repository

Secure Development Guide

Release 8.0

E98318-01

February 2019

The *Secure Development Guide* provides an overview of the security options for customers who will use Healthcare Data Repository (HDR) user database accounts and middle tier WebLogic user accounts to access the HDR APIs. Note that the set of recommendations in this document is not exhaustive and that no guarantee is given that implementing all the suggestions in this document provides sufficient protection for all security threats. The reason for this disclaimer is that you cannot delegate responsibility for secure application development to a third party or a single document. This document is to help developers be aware of the security tools and features that they can use to implement application security. This document does not replace a formal code review process.

Guidelines are presented here to assist in mitigating common security risks when customers are using the HDR APIs. The Open Web Application Security Project (OWASP) publishes the OWASP Top 10 to identify some of the most critical application security risks. This document briefly describes each Top 10 risk, provides the HDR mitigation strategies, and encourages our users to extend these strategies to secure their own applications and environments that use our APIs. Some of the web-specific Top 10 items don't apply to HDR; these are marked as **Not Applicable**.

1 OWASP Top 10 Security Vulnerabilities 2013

This paper discusses the practices and strategies used by the HDR application to mitigate risks posed by the security vulnerabilities documented in the OWASP Top 10 – 2013. Customers using the HDR APIs should be aware of and protect against these threats as well.

For the OWASP Foundation's description of the OWASP Top 10 Application Security Risks see the following document: https://www.owasp.org/images/f/f8/OWASP_Top_10_-_2013.pdf.

1.1 Injection

SQL Injection can occur when untrusted data is used in a command or query. If an attacker sends hostile data, this can result in executing harmful commands or in unauthorized access of data.

The HDR APIs accept query parameters which could potentially contain hostile data. HDR code adheres to the recommended standards to avoid SQL Injection possibilities by using bind variables and by checking and validating the user input.

Customers using the HDR APIs to build applications should code carefully, using proper data validations and checking user input where needed.

1.2 Broken Authentication and Session Management

The risk of insecure login credentials or session IDs increases with custom built authentication schemes.

The HDR APIs are secured using the standard WebLogic authentication mechanism. Follow the WebLogic recommendations on creating and configuring a suitable authentication provider where the HDR WebLogic user accounts will be stored. Refer to https://docs.oracle.com/middleware/12213/wls/SECMG/toc.htm.

Avoid building a custom authentication provider for using the HDR APIs. Use strong passwords and maintain security of account credentials.

1.3 Cross-Site Scripting (XSS)

Not applicable for access to HDR APIs.

1.4 Insecure Direct Object References

Direct object references are insecure when an attacker is able to substitute a reference to an internal application object with a reference to another object that the user is not authorized to access and the application doesn't verify the user's authorization, but permits the unauthorized access.

User authorization and direct object references must be handled in the application code that uses the HDR APIs. All HDR APIs are accessible to any authenticated user. This user is an internal user and not the same as the users that log in to the application developed using the HDR APIs.

In software using the HDR APIs, a direct object reference from an untrusted source should verify the user authorization to access the object.

1.5 Security Misconfiguration

Attackers can take advantage when the security configuration is incorrect or incomplete and obtain unauthorized access to an application. The entire technology stack must be configured properly, and processes should be in place to detect misconfigurations—missing patches, accounts with default passwords, insecure settings in frameworks or libraries, etc.

Since HDR is an on-premise application, it must be deployed in a secure WebLogic instance.

Ensure that your WebLogic configuration where HDR is deployed does not contain opportunities for an attacker to gain unauthorized access to the system. Take care to secure default accounts, files or directories, servers, the network, and other data access channels of the HDR deployment and APIs.

1.6 Sensitive Data Exposure

Attackers may obtain unauthorized access to poorly protected sensitive data. Caution should be used to hide sensitive information from unauthorized users.

User authorization is not part of the HDR application itself and it is strongly recommended that the applications developed using the HDR APIs have their own user management, authorization and access control mechanisms to allow controlled access to data stored in HDR.

1.7 Missing Function Level Access Control

In some applications, the UI controls function-level access by exposing only the functionality for which access has been granted and hiding functionality where the user has not been granted access. If there is no database-level access control, then an attacker, who has partial access to the application, might gain access to an unauthorized function.

The software that uses the HDR APIs should properly enforce access privileges for application functions at the level of the business logic. The default behavior in an application should be to deny access to application functions unless the access is granted explicitly.

1.8 Cross-Site Request Forgery (CSRF)

Not applicable for access to HDR APIs.

1.9 Using Components with Known Vulnerabilities

If an older version of a component with a known vulnerability is deployed in an environment, then an attacker who is aware of the vulnerability could take advantage and obtain unauthorized access.

HDR is built on a technology stack where patches and new releases offer improvements, including security-related modifications. The HDR application is an on-premise application and users should download and apply recommended security patches for the respective versions of components like WebLogic server and Oracle Database.

HDR users should follow the same policy of applying patches and updating to the latest versions of components being used, especially if security vulnerabilities have been reported in older versions.

1.10 Non-Validated Redirects and Forwards

Not applicable for access to HDR APIs.

2 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.

Oracle Healthcare Data Repository Secure Development Guide, Release 8.0
E98318-01