Hewlett Packard Enterprise

HPE Security Fortify Standalone Report Generator

OWASP Top 10 2013

tasy-agent

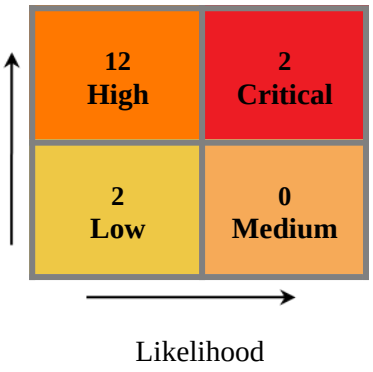
FORTIFY

Table of Contents

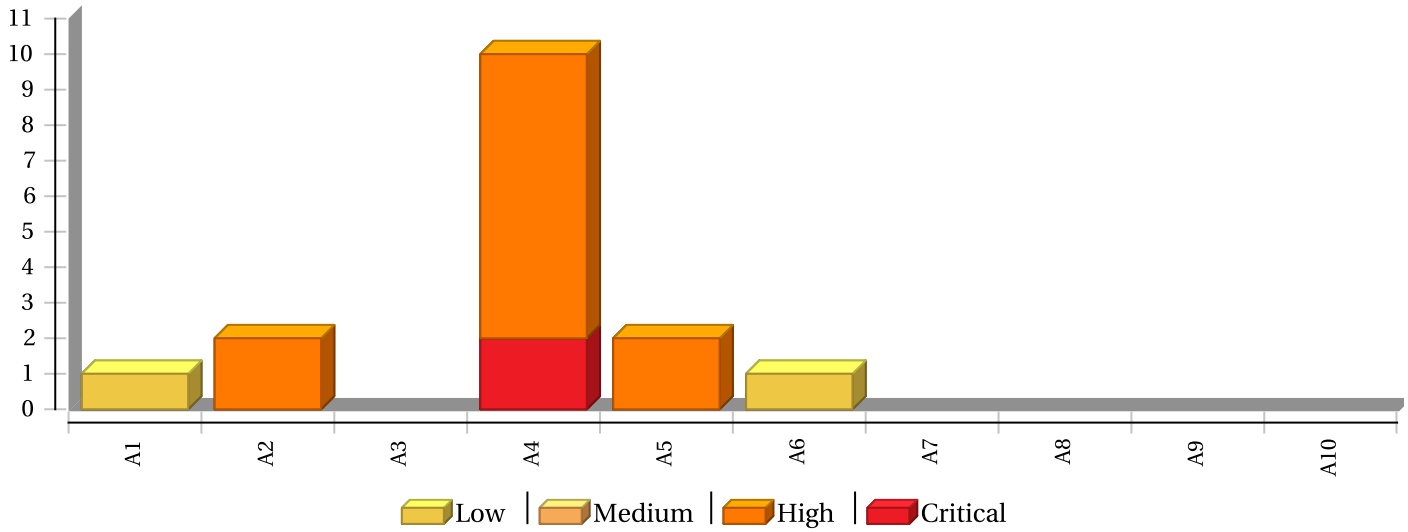
Executive Summary

Project Name:	tasy-agent
Project Version:	
SCA:	Results Present
WebInspect:	Results Not Present
SecurityScope:	Results Not Present
Other:	Results Not Present

Issues by Priority



Issues by OWASP Top 10 2013 Categories



* The detailed sections following the Executive Summary contain specifics.

Project Description

This section provides an overview of the HPE Security Fortify scan engines used for this project, as well as the project meta-information.

SCA

Date of Last Analysis:	Feb 1, 2018, 7:44 AM	**Engine Version:**	17.10.0156
Host Name:	srv-sec-protex.whebdc.com.br	**Certification:**	VALID
Number of Files:	76	**Lines of Code:**	3,524

Issue BreakDown

The following table summarizes the number of issues identified across the different OWASP Top 10 2013 categories and broken down by Fortify Priority Order.

	Fortify Priority				Total Issues
	Critical	High	Medium	Low	
A1 Injection	0	0	0	1	1
A2 Broken Authentication and Session Management	0	2	0	0	2
A3 Cross-Site Scripting (XSS)	0	0	0	0	0
A4 Insecure Direct Object References	2	8	0	0	10
A5 Security Misconfiguration	0	2	0	0	2
A6 Sensitive Data Exposure	0	0	0	1	1
A7 Missing Function Level Access Control	0	0	0	0	0
A8 Cross-Site Request Forgery (CSRF)	0	0	0	0	0
A9 Using Components with Known Vulnerabilities	0	0	0	0	0
A10 Unvalidated Redirects and Forwards	0	0	0	0	0

NOTE:
1. Reported issues in the above table may violate more than one OWASP Top 10 2013 category. As such, the same issue may appear in more than one row. The total number of unique vulnerabilities are reported in the Executive Summary table.

Issue Details

Below is an enumeration of all issues found in the project. The issues are organized by OWASP Top 10 2013, Fortify Priority Order, and vulnerability category. The issues are then further broken down by the package, namespace, or location in which they occur. Issues reported at the same line number with the same category originate from different taint sources.

A1 Injection

Injection flaws, particularly SQL injection, are common in web applications. Injection occurs when user-supplied data is sent to an interpreter as part of a command or query. The attacker's hostile data tricks the interpreter into executing unintended commands or changing data.

Command Injection		Low
Package: com.philips.tasy.agent.client.adminservices		
Location	**Analysis Info**	**Analyzer**
client-services/src/main/java/com/philips/tasy/agent/client/adminservices/UpdateService.java: 208	**Sink:** ProcessBuilder(0) **Enclosing Method:** updateClient() **Source:**	SCA

A2 Broken Authentication and Session Management

Account credentials and session tokens are often not properly protected. Attackers compromise passwords, keys, or authentication tokens to assume other users' identities.

Often Misused: Authentication		High
Package: com.philips.tasy.agent.commons.http		
Location	**Analysis Info**	**Analyzer**
commons/src/main/java/com/philips/tasy/agent/commons/http/ServerMonitor.java:35	**Sink:** getByName() **Enclosing Method:** stopServer() **Source:**	SCA
commons/src/main/java/com/philips/tasy/agent/commons/http/ServerMonitor.java:53	**Sink:** getByName() **Enclosing Method:** startUp() **Source:**	SCA

A3 Cross-Site Scripting (XSS)

XSS flaws occur whenever an application takes user supplied data and sends it to a web browser without first validating or encoding that content. XSS allows attackers to execute script in the victim's browser which can hijack user sessions, deface web sites, possibly introduce worms, etc.

No Issues

A4 Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization.

Path Manipulation	Critical

Package: com.philips.tasy.agent.commons

Location	Analysis Info	Analyzer
commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:28	**Sink:** java.io.File.File() **Enclosing Method:** checkJndi() **Source:** javax.naming.Context.lookup() **from** com.philips.tasy.agent.commons.HomeDirLocator.checkJndi() **In** commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:26	SCA
commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:33	**Sink:** java.io.File.File() **Enclosing Method:** checkJndi() **Source:** javax.naming.InitialContext.lookup() **from** com.philips.tasy.agent.commons.HomeDirLocator.checkJndi() **In** commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:31	SCA

Path Manipulation	High

Package: com.philips.tasy.agent.client.adminservices

Location	Analysis Info	Analyzer
client-services/src/main/java/com/philips/tasy/agent/client/adminservices/UpdateService.java:206	**Sink:** java.nio.file.Paths.get() **Enclosing Method:** updateClient() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.client.adminservices.UpdateService.updateClient() **In** client-services/src/main/java/com/philips/tasy/agent/client/adminservices/UpdateService.java:206	SCA

Package: com.philips.tasy.agent.client.core.server

Location	Analysis Info	Analyzer
client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClientPaths.java:31	**Sink:** java.io.File.File() **Enclosing Method:** updateFolder() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.client.core.server.ClientPaths.getHomeDir() **In** client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClientPaths.java:51	SCA
client-core/src/main/java/com/philips/tasy/agent/client/core/server/ConfigurationProvider.java:56	**Sink:** java.io.File.File() **Enclosing Method:** getConfigFile() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.client.core.server.ClientPaths.getHomeDir() **In** client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClientPaths.java:51	SCA
client-core/src/main/java/com/philips/tasy/agent/client/core/server/ConfigurationProvider.java:60	**Sink:** java.io.File.File() **Enclosing Method:** getConfigFile() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.client.core.server.ClientPaths.getHomeDir() **In** client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClientPaths.java:51	SCA

A4 Insecure Direct Object References

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, database record, or key, as a URL or form parameter. Attackers can manipulate those references to access other objects without authorization.

Path Manipulation	High

Package: com.philips.tasy.agent.client.core.server

Location	Analysis Info	Analyzer
client-core/src/main/java/com/philips/tasy/agent/client/core/server/TasyAgentConfiguration.java:21	**Sink:** java.io.File.File() **Enclosing Method:** TasyAgentConfiguration() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.client.core.server.ClientPaths.baseTempFolder() **In** client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClientPaths.java:59	SCA
client-core/src/main/java/com/philips/tasy/agent/client/core/server/TasyAgentConfiguration.java:22	**Sink:** java.io.File.File() **Enclosing Method:** TasyAgentConfiguration() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.client.core.server.ClientPaths.getHomeDir() **In** client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClientPaths.java:51	SCA

Package: com.philips.tasy.agent.commons

Location	Analysis Info	Analyzer
commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:48	**Sink:** java.io.File.File() **Enclosing Method:** checkSysProps() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.commons.HomeDirLocator.checkSysProps() **In** commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:46	SCA
commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:85	**Sink:** java.io.File.File() **Enclosing Method:** fromUserHome() **Source:** java.lang.System.getProperty() **from** com.philips.tasy.agent.commons.HomeDirLocator.fromUserHome() **In** commons/src/main/java/com/philips/tasy/agent/commons/HomeDirLocator.java:85	SCA

A5 Security Misconfiguration

Having a strong server configuration standard is critical to a secure web application. Servers have many configuration options that affect security and many are not secure out of the box.

Access Specifier Manipulation		High
Package: com.philips.tasy.agent.client.core.server		
Location	**Analysis Info**	**Analyzer**
client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClassLoaderCleaner.java:57	**Sink:** setAccessible() **Enclosing Method:** getValueFromField() **Source:**	SCA
client-core/src/main/java/com/philips/tasy/agent/client/core/server/ClassLoaderCleaner.java:202	**Sink:** setAccessible() **Enclosing Method:** finalizeNativeLibs() **Source:**	SCA

A6 Sensitive Data Exposure

Many web applications do not properly protect sensitive data, such as credit cards, tax IDs, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes.

Weak Cryptographic Hash: Missing Required Step		Low
Package: com.philips.tasy.agent.commons		
Location	**Analysis Info**	**Analyzer**
commons/src/main/java/com/philips/tasy/agent/commons/Hash.java:49	**Sink:** digest.digest() : Cryptographic hash finalized without update **Enclosing Method:** checksum() **Source:**	SCA

A7 Missing Function Level Access Control

Most web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed.

No Issues

A8 Cross-Site Request Forgery (CSRF)

CSRF attacks force an authenticated victim's browser to send an unauthenticated request to a vulnerable web application, which then performs unauthorized action on behalf of the attacker. CSRF can be as powerful as the web application that it targets.

No Issues

A9 Using Components with Known Vulnerabilities

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.

No Issues

A10 Unvalidated Redirects and Forwards

Redirects allow web applications to direct users to different pages within the same application or to external sites. Attackers can utilize open redirects to trick users into visiting a URL to a trusted site and redirecting them to a malicious site. Open redirects are often abused as part of phishing scams to harvest sensitive end-user data.

No Issues