

FOD Quick start

The screenshot shows the FOD dashboard interface. At the top, there is a blue header bar with the FOD logo, navigation links for APPLICATIONS, DASHBOARD, REPORTS, and ADMINISTRATION, and user information for ZACH. Below the header, the main dashboard area has a title "Demo FPR Upload". It features three main sections: "Policy Compliance" (with a star rating of 1/5 and buttons for FAIL and VIEW), "Issues In Production" (showing 20 CRITICAL, 7 HIGH, 0 MEDIUM, and 0 LOW issues), and "Security Status" (listing APP DEFENDER as DISABLED, MONITORING as DISABLED, and NETWORK as NOT STARTED). On the left side, there is a vertical sidebar with icons for Applications, Dashboards, Reports, Administration, and Help.

Demo FPR Upload

Policy Compliance

FAIL [VIEW](#)

Issues In Production

CRITICAL	HIGH	MEDIUM	LOW
20	7	0	0

Security Status

APP DEFENDER	MONITORING	NETWORK
DISABLED	DISABLED	NOT STARTED

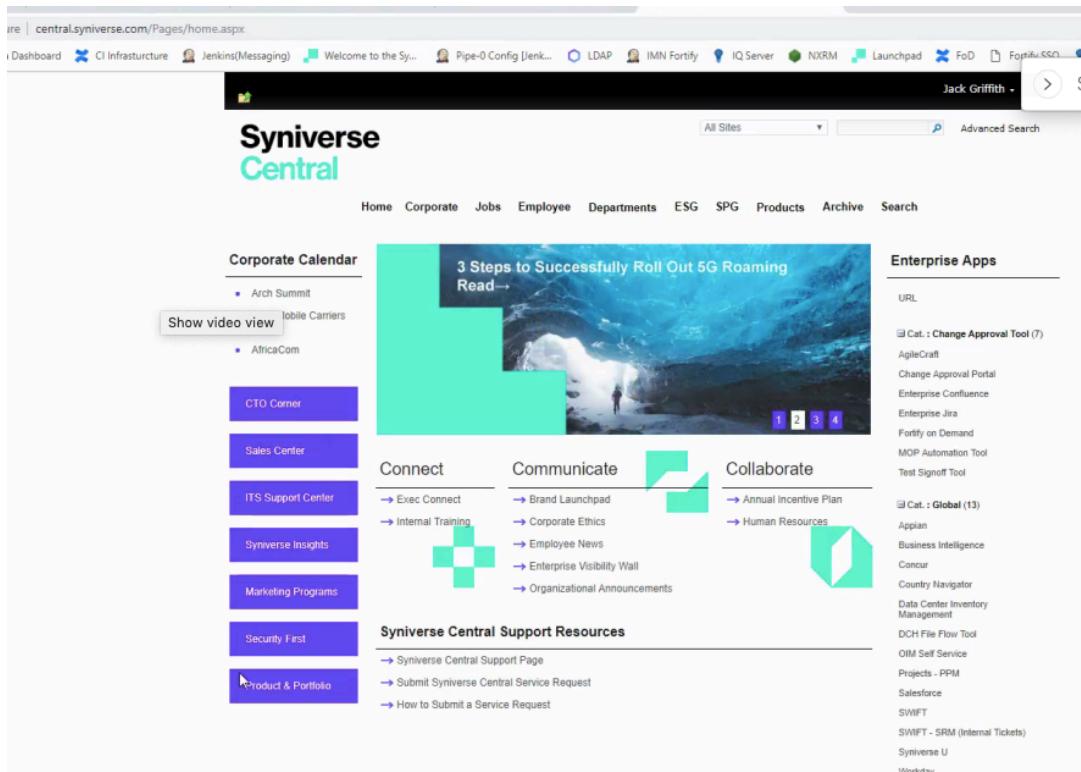
Agenda

1. How to log in
2. How to upload code
3. How to initiate a scan
4. How to check progress of a scan
5. How to view results (as a whole set)
6. How to sort, prioritize, parse results
7. How to dive deeper into individual results
8. How to create reports
9. How to use the Security Assistant
10. How to access Secure Code Warrior

How to log in

NOTE: You should have received an email with your login information for user, password and tenant.

Open Web Browser and go to <https://central.syniverse.com>



How to log in - 2

Click Fortify on Demand Link

The screenshot shows the Syniverse Central homepage. At the top, there is a navigation bar with links like Dashboard, CI Infrastructure, Jenkins(Messaging), Welcome to the Sy..., Pipe-0 Config [Jen...], LDAP, IMN Fortify, IQ Server, NXRM, Launchpad, FoD, and Fortify SSO. A user profile for "Jack Griffith" is visible. Below the navigation bar, the main content area features the Syniverse Central logo and a search bar. On the left, there's a "Corporate Calendar" section with events like Arch Summit and AfricaCom, and a "Show video view" button. To the right of the calendar is a large banner titled "3 Steps to Successfully Roll Out 5G Roaming" with a "Read →" button and a small image of a person in a snowy landscape. Below the banner are three columns: "Connect" (with icons for Exec Connect and Internal Training), "Communicate" (with icons for Brand Launchpad, Corporate Ethics, Employee News, Enterprise Visibility Wall, and Organizational Announcements), and "Collaborate" (with icons for Annual Incentive Plan and Human Resources). At the bottom, there's a "Syniverse Central Support Resources" section with links to the Support Page, Service Request submission, and How to Submit a Service Request. On the far right, there's a sidebar titled "Enterprise Apps" listing various tools and services.

central.syniverse.com/Pages/home.aspx

Dashboard CI Infrastructure Jenkins(Messaging) Welcome to the Sy... Pipe-0 Config [Jen... LDAP IMN Fortify IQ Server NXRM Launchpad FoD Fortify SSO

Jack Griffith

Syniverse Central

All Sites Advanced Search

Home Corporate Jobs Employee Departments ESG SPG Products Archive Search

Corporate Calendar

- Arch Summit
- Show video view
- AfricaCom

CTO Corner

Sales Center

ITS Support Center

Syniverse Insights

Marketing Programs

Security First

Product & Portfolio

3 Steps to Successfully Roll Out 5G Roaming
Read →

1 2 3 4

Connect

- Exec Connect
- Internal Training

Communicate

- Brand Launchpad
- Corporate Ethics
- Employee News
- Enterprise Visibility Wall
- Organizational Announcements

Collaborate

- Annual Incentive Plan
- Human Resources

Syniverse Central Support Resources

- Syniverse Central Support Page
- Submit Syniverse Central Service Request
- How to Submit a Service Request

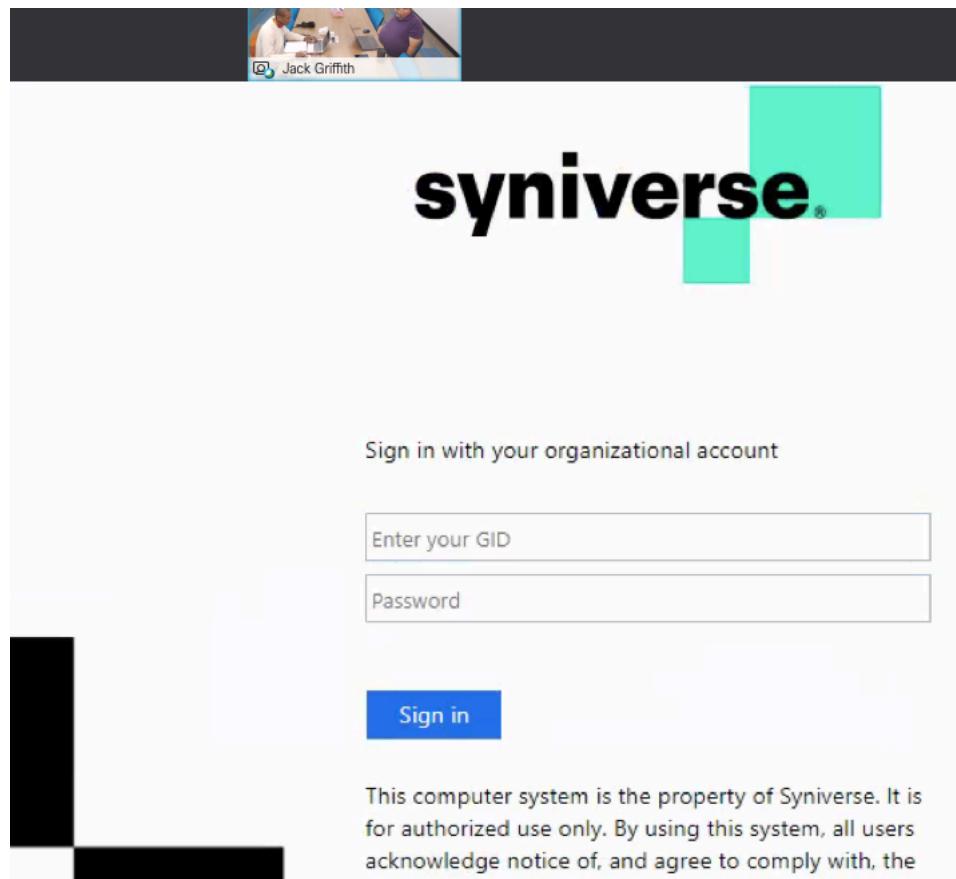
Enterprise Apps

URL

- Cat. : Change Approval Tool (7)
 - AgileCraft
 - Change Approval Portal
 - Enterprise Confluence
 - Enterprise Jira
 - Fortify on Demand
 - MOP Automation Tool
 - Test Signoff Tool
- Cat. : Global (13)
 - Appian
 - Business Intelligence
 - Concur
 - Country Navigator
 - Data Center Inventory Management
 - DCH File Flow Tool
 - QIM Self Service
 - Projects - PPM
 - Salesforce
 - SWIFT
 - SWIFT - SRM (Internal Tickets)
 - Syniverse U
 - Workday

How to log in - 3

Login with your SSO



Understanding the Dashboard

Main Dashboard Screen

The screenshot shows the main dashboard interface for managing applications. The top navigation bar includes links for APPLICATIONS, DASHBOARD, REPORTS, and ADMINISTRATION, along with user information for ZACH. The left sidebar features icons for navigation, applications, releases, and scans. The central area displays a list of managed applications:

NAME	PRODUCTION RISK & POLICY COMPLIANCE	SCAN
Demo FPR Upload 2 RELEASES Business Criticality: HIGH	<div style="display: flex; align-items: center;"><div style="flex-grow: 1;">CRITICAL: 20</div><div style="flex-grow: 1;">HIGH: 7</div><div style="flex-grow: 1;">MEDIUM: 0</div><div style="flex-grow: 1;">LOW: 0</div></div>	STATIC:
FODDemoMissingJar 2 RELEASES Business Criticality: HIGH	pre-production	STATIC:

A yellow arrow points to the star rating icon for the first application. The right side of the screen contains a sidebar with filters for SORT (Application Name A to Z), MOST RECENT CHANGE (Release Failing Security Policy, New Static Vulnerabilities Detected), and PASS/FAIL.

Preparing Java Zip for Upload

For Java applications, package the compiled and source code files for upload to Fortify on Demand as follows.

Preparing Compiled Files (Required)

Compile all files in debug mode. If they are not, the assessment will still be completed, but the results will not include line numbers of issues.

Ensure that all third-party JAR files are included. This prevents missing dependency issues from appearing during assessment.

Web application: Package in a WAR or EAR file. You can package multiple WAR files as an EAR file.

Non-web application: Ensure that there are no precompiled JSPs.

Package all application files in one zip file.

Preparing Java Zip for Upload -2

```
#!/bin/sh
#
# Simple Java ZIP Files and Jars for upload to FOD.
#
if [ "$#" -gt 1 ];
Then
APP_SOURCE_DIR=$1
APP_JAR_DIR=$2

mkdir tmp_fodupload

for appfiles in `find ${APP_SOURCE_DIR} -type f -exec file {} \; | grep "ASCII text" | cut -d: -f1`  
do
cp $appfiles tmp_fodupload
done

for jar_files in `find ${APP_JAR_DIR} -name "*.jar" `
do
cp ${jar_files} tmp_fodupload
done

zip -r tmp_fodupload.zip tmp_fodupload

else

echo "usage fodZipJava <App Source Dir> <App Jar Dir>"  
fi
```

Start Scan - 1

Select Start Scan

The screenshot shows a software interface with a blue header bar containing 'APPLICATIONS', 'DASHBOARD', 'REPORTS', and 'ADMINISTRATION'. A search bar with a count of '70' and a user dropdown for 'ZACH' are also in the header. On the left, there's a vertical sidebar with icons for applications, dashboard, reports, administration, PDF, settings, users, and logs.

The main area displays 'Demo FPR Upload' information:

- Policy Compliance:** FAIL, VIEW
- Issues In Production:** CRITICAL: 20, HIGH: 7, MEDIUM: 0, LOW: 0
- Security Status:** APP DEFENDER: DISABLED, MONITORING: DISABLED, NETWORK: NOT STARTED

Below this is a 'Releases' section:

- Search bar: Search Text
- Buttons: + NEW RELEASE, Filter icon
- Table headers: Display: 25, 50, 100, START SCAN, RELEASE, SDLC STATUS, POLICY COMPLIANCE, # ISSUES, STATIC, DYNAMIC, LAST COMPLETE
- Data row:
 - START SCAN: **START SCAN ▶** (highlighted with a yellow arrow)
 - RELEASE: FAIL
 - SDLC STATUS: **FAIL**
 - POLICY COMPLIANCE: ★★★★★
 - # ISSUES: CRITICAL: 20, HIGH: 7, MEDIUM: 0, LOW: 0
 - STATIC: ✓
 - DYNAMIC: -
 - LAST COMPLETE: 2019/02/2

Start Scan - 2

Click Start Scan

NOTE: The DevOps team has already setup this information.

 **Static Scan Setup** 

Static Scan Details 9975 Unit(s) Available

Assessment Type
Static Assessment - Subscription (4 Units)

Source or Compiled Code/Files 
Manual Upload

The service level objective (SLO) for this assessment is 1 business day(s) Pacific Standard Time 

Technology Stack
JAVA/J2EE

Language Level
(Choose One)

Audit Preference
Manual

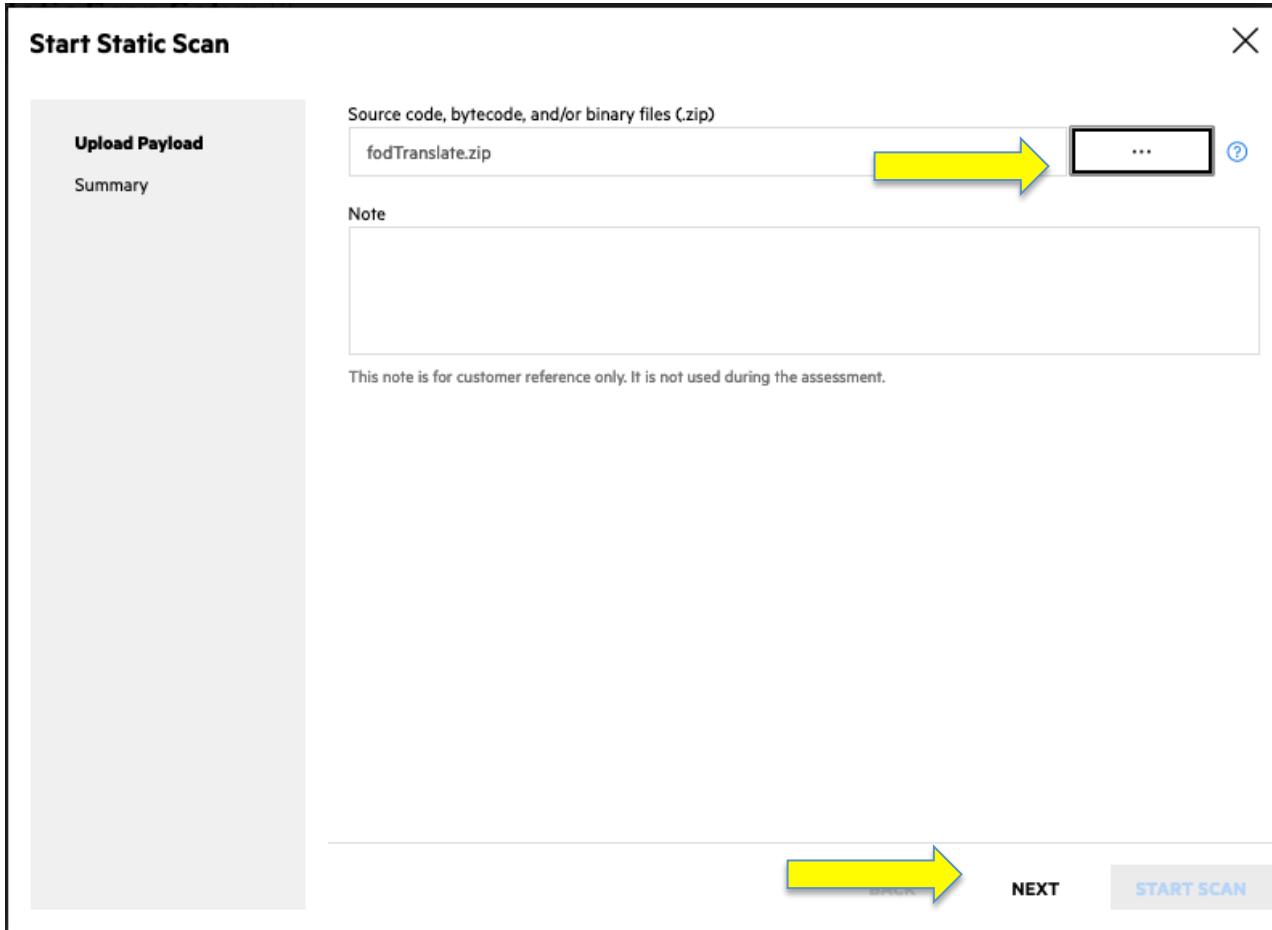
Audit preference determines whether a security expert will manually review the scan results for overall quality and to remove false positives. Selecting Automated audit will automatically suppress new issues identified as false positives by Fortify on Demand Scan Analytics with high confidence and publish the results without human review, which can reduce the turnaround time. We recommend a manual audit for the first scan of an application or release with major changes, with subsequent scans using automated audit for shorter turnaround times. False positive challenges can only be requested for issues introduced during a scan with manual audit.

Build Server Integration Token 

No token could be generated based on current criteria

Start Scan - 3

Click ... to Upload Zip File and then click next



Start Scan - 4

Click Start Scan

Start Static Scan

✓ Upload Payload

Summary

Scan Details

Assessment Type
Static Assessment - Subscription (4 Units)
The service level objective (SLO) for this assessment is 1 business day(s) Pacific Standard Time [?](#)

Technology Stack JAVA/J2EE	Audit Preference Manual
Language Level 1.8	Include third-party libraries for static security assessment No
Open Source Component Analysis No	
Remediation Scan No	

Payload Details

Source or Compiled Code/Files
Manual Upload

Source code, bytecode, and/or binary files (.zip)
fodTranslate.zip

BACK  **START SCAN**

Start Scan - 5

Scan in Progress

Demo FPR Upload > Try2

Static - Scan In Progress

 Static Scan Setup [?](#)

Static Scan Details

Assessment Type: Static Assessment - Subscription (Ends 12/20/2019)

The service level objective (SLO) for this assessment is 1 business day(s) Pacific Standard Time [?](#)

Source or Compiled Code/Files: Manual Upload

Technology Stack: JAVA/J2EE

9971 Unit(s) Available

SAVE

Check Status of Scan

https://sandbox.fortify.com/Applications?rpp=25&sort=application&sortdir=Asc

APPLICATIONS DASHBOARD REPORTS ADMINISTRATION

Your Applications

Managed 5 Discovered 0 Ignored 0

5 found

NAME	PRODUCTION RISK & POLICY COMPLIANCE 1					SCAN
Demo FPR Upload 2 RELEASES Business Criticality: HIGH	FAIL	CRITICAL	HIGH	MEDIUM	LOW	STATIC
FODDemoMissingJar 2 RELEASES Business Criticality: HIGH	★★★★★	20	7	0	0	STATIC
Java 11 MBS Example 1 RELEASES Business Criticality: HIGH			pre-production			STATIC

Check Status of Scan

The screenshot shows the Fortify application interface. The top navigation bar includes links for APPLICATIONS, DASHBOARD, REPORTS, and ADMINISTRATION, along with a search icon, a notifications badge (72), and user profile icons.

The left sidebar contains icons and labels for OVERVIEW, ISSUES, SCANS, MONITORING, REPORTS, SETTINGS, ACCESS, and AUDIT TEMPLATE. The MONITORING section is currently selected, displaying a table of events.

A yellow arrow points to the "EVENT LOG" link in the sidebar, which is highlighted in blue. The table below lists events with columns for DATE, TYPE, USER, and NOTES.

DATE	TYPE	USER	NOTES
5:42 AM	Static Scan Rejected	zachlewis	notes = "Reason: Missing JAR/CLASS or JAVA files, Notes: Java payloads must contain .java, .jsp, .jar, or .class files"
5:42 AM	Analysis Payload Uploaded	zachlewis	uploadedBy = "zachary.lewis@microfocus.com"
5:39 AM	User Start Scan Static	zachlewis	scanId = "1883", notes = ""
7:24 AM	Static Scan Completed	zachlewis	scanId = "1876", projectVersionId = "4287", notes = ""
7:09 AM	User Start Scan Static	zachlewis	scanId = "1876", notes = ""
5:46 AM	Release Created	zachlewis	ReleaseName = "Try2", SDLCStatus = "1", ReleaseRedirectUri = "https://sandbox.fortify.com/redirect/Releases/4287", ApplicationName = "Demo FPR Upload", ApplicationWeb/Thick-Client, ApplicationBusinessCriticality = "1", ApplicationRedirectUri = "https://sandbox.fortify.com/redirect/Applications/4054"
5:32 AM	Static Scan Completed	zachlewis	scanId = "1875", projectVersionId = "4286", notes = ""
5:27 AM	User Start Scan Static	zachlewis	scanId = "1875", notes = ""
2:08 AM	Static Scan Rejected	zachlewis	notes = "Reason: Missing JAR/CLASS or JAVA files, Notes: Java payloads must contain .java, .jsp, .jar, or .class files"
2:03 AM	User Start Scan Static	zachlewis	scanId = "1874", notes = ""

Display: 25

FOD Reviewing Results

Understanding the Sink and what needs to be done.

The screenshot shows a software interface with a blue header bar containing the following navigation items: APPLICATIONS, DASHBOARD, REPORTS, and ADMINISTRATION. On the far right of the header, there is a search icon, a notifications badge with the number 70, a help icon, a user dropdown menu labeled ZACH, and a dropdown arrow.

The main content area has a light gray background and features three main sections:

- Policy Compliance:** Displays a star rating icon with four stars filled and one empty, followed by the word "FAIL" in red and a "VIEW" button in blue.
- Issues In Production:** A summary of critical, high, medium, and low issues. The counts are: CRITICAL (20), HIGH (7), MEDIUM (0), and LOW (0). These counts are displayed in colored circles: red for Critical, orange for High, light gray for Medium, and light gray for Low.
- Security Status:** Shows the status of three components: APP DEFENDER (disabled), MONITORING (disabled), and NETWORK (not started). Each component has a circular icon with a minus sign and a descriptive label below it.

A vertical sidebar on the left contains icons for navigation: three horizontal lines, a person, a globe, a magnifying glass, and a left arrow.

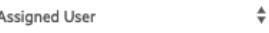
FOD Reviewing Results - 2

Demo FPR Upload

Application Issues

SDLC STATUS: PRODUCTION 

20 7 0 0 27     

Group By 

✓ (Not Set) 
 sink.asp : 17
 sink.asp : 17
 command-injection.ba...
 command-injection.ba...
 command-injection.ba...
 sample.js : 2

70524 basic/vbscript/sink.asp : 17       

Vulnerability Recommendations Code Diagram More Evidence 

Summary

Line 17 of `sink.asp` invokes a SQL query built using input coming from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands. Constructing a dynamic SQL statement with input coming from an untrusted source could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

Explanation

SQL injection errors occur when:

1. Data enters a program from an untrusted source.

expand all | collapse all 

SEARCH TEXT  SHOW  

expand all | collapse all 

✓ CANNED QUERIES 

> SEVERITY
> CATEGORY
> SINK
> SOURCE
> OWASP 2017
> ISSUE AGE

FOD Reviewing Results - 3

70524 basic/vbscript/sink.asp : 17

Try2

Critical

SQL Injection

SMART FIX

Vulnerability

Recommendations

Code

Diagram

More Evidence ▾

History

Summary

Line **17** of **sink.asp** invokes a SQL query built using input coming from an untrusted source. This call could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands. Constructing a dynamic SQL statement with input coming from an untrusted source could allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

Explanation

SQL injection errors occur when:

1. Data enters a program from an untrusted source.

FOD Reviewing Results - 4

Try2

Critical

SQL Injection



SMART FIX

Vulnerability

Recommendations

Code

Diagram

More Evidence ▾

History



FOD Reviewing Results - 5

70524 basic/vbscript/sink.asp : 17

Try2 Critical SQL Injection SMART FIX

Vulnerability Recommendations **Code** Diagram More Evidence History

Path 1 of 1 Word Wrap Stacked View Inline Analysis Trace

source.asp:6 - Read request.cookies['hello']

source.asp:6 - Assignment to sql1

source.asp:9 - mysqlstuff(1)

sink.asp:17 - open(0)

```
1 <!-- go include the sink definition, so it's available on this page. --
2 <!-- #include file="sink.asp" -->
3 <%
4     Set pwd="bar"
5
6     Set sql1="SELECT companyname FROM " & Request.Cookies("hello")
7     Set sql2=Request.QueryString("foo")
8
9     MySqlStuff pwd, sql1, sql2
10
11 %>
12
```

FOD Reviewing Results - 6

Demo FPR Upload > Try2

Release Issues

70524 basic/vbscript/sink.asp : 17

Try2

Critical

SQL Injection



SMART FIX

Vulnerability

Recommendations

Code

Diagram

More Evidence ▾

History

Recommendation

The root cause of a SQL injection vulnerability is the ability of an attacker to change context in the SQL query, causing a value that the programmer intended to be interpreted as data to be interpreted as a command instead. When a SQL query is constructed, the programmer knows what should be interpreted as part of the command and what should be interpreted as data. Parameterized SQL statements can enforce this behavior by disallowing data-directed context changes and preventing nearly all SQL injection attacks. Parameterized SQL

the user to select from this set of application-controlled values.

Tips

1. A common mistake is to use parameterized SQL statements that are constructed by concatenating user-controlled strings. Of course, this defeats the purpose of using parameterized SQL statements. If you are not certain that the strings used to form parameterized statements are constants controlled by the application, do not assume that they are safe because they are not being executed directly as SQL strings. Thoroughly investigate all uses of user-controlled strings in SQL statements and verify that none can be used to modify the meaning of the query.

NEW Interactive Training

Through our partnership with Secure Code Warrior, try a short, hands-on challenge where sample software code needs to be analyzed for SQL Injection security weaknesses. Once identified, you can then modify the code to remediate or mitigate the SQL Injection vulnerability. Fortify on Demand does not share your personal or organization's information with Secure Code Warrior.

[LAUNCH TRAINING](#)

References

1. SQL Injection Attacks by Example, S. J. Friedl, <http://www.unixwiz.net/techtips/sql-injection.html>
2. Stop SQL Injection Attacks Before They Stop You, P. Litwin
3. SQL Injection and Oracle, Part One, P. Finnigan, <http://www.securityfocus.com/infocus/1644>
4. Writing Secure Code, Second Edition, M. Howard, D. LeBlanc
5. CWE ID 89, Standards Mapping - Common Weakness Enumeration

FOD Reviewing Results - 7

Common Weakness Enumeration: CWE

<https://cwe.mitre.org/> ▾

Apr 3, 2018 - **Common Weakness Enumeration (CWE)** is a list of software weaknesses.

[About CWE](#) · [CWE List](#) · [Search](#) · [CWE - Reports](#)

CWE - CWE-89: Improper Neutralization of Special Elements used in ...

<https://cwe.mitre.org> › [CWE List](#) ▾

CWE-89: Improper Neutralization of Special Elements used in an SQL Command The programmer may have skipped any input validation on \$id under the ...

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy, i.e., use a whitelist of acceptable inputs that strictly conform to specifications. Reject any input that does not strictly conform to specifications, or transform it into something that does.

MemberOf	 1005	7PK - Input Validation and Representation
MemberOf	 1027	OWASP Top Ten 2017 Category A1 - Injection

FOD Reporting

The screenshot shows the 'Your Reports' page. At the top, there is a navigation bar with links for APPLICATIONS, DASHBOARD, REPORTS (which is the active tab), and ADMINISTRATION. On the right side of the header, there is a search bar, a notifications icon (72), and a user account for ZACH. Below the header, the page title 'Your Reports' is displayed, followed by a message '4 found'. To the left of the main content area, there is a sidebar with icons for PDF, CSV, and other report formats. The main content area displays a table with columns: REPORT NAME, APPLICATION, RELEASE, CREATED DATE, REPORT TEMPLATE, LANGUAGE, STATUS, and CREATED BY. One row is visible, showing 'd1' as the report name, 'Java App Demo' as the application, 'v1' as the release, '2019/02/26 12:53:55 PM' as the created date, 'Static Comprehensive' as the report template, 'English' as the language, 'Failed' as the status, and 'zachlewis' as the created by user. There are also buttons for 'Search Text', 'Display' (set to 25), and '+ NEW REPORT'.

The screenshot shows the 'Create Report' dialog box. The title bar says 'Create Report' with a close button. On the left, there is a sidebar with a checked checkbox 'Select Release' and three tabs: 'Report Details' (selected), 'Report Template', and 'Summary'. The main area contains fields for 'Report Name' (with a yellow arrow pointing to it), 'Notes' (a text area), and 'File Type' (a dropdown menu set to 'PDF'). At the bottom right, there are 'NEXT' and 'GENERATE' buttons, with a yellow arrow pointing to the 'NEXT' button.

FOD Reporting - 2

Create Report

- ✓ Select Release
- ✓ Report Details
- Report Template**
- Summary

Report Template

- [FISMA Compliance](#)
- [Hybrid Comprehensive](#)
- [Hybrid Issue Detail](#)
- [Hybrid Summary](#)
- [PCI 2.0 DSS Compliance](#)
- [PCI 3.0 DSS Compliance](#)
- [PCI 3.1 DSS Compliance](#)
- [PCI 3.2 DSS Compliance](#)
- [STIG 3.9 Compliance](#)
- [STIG 4.1 Compliance](#)
- [STIG 4.3 Compliance](#)
- [Static Analysis Trace](#)
- [Static Comprehensive](#)
- [Static Issue Detail](#)
- [Static Summary](#)

BACK NEXTGENER

FOD Reporting - 3

Create Report

X

- ✓ Select Release
- ✓ Report Details
- ✓ Report Template

Summary

Report Name
zzz

Notes

File Type
PDF

Application
Java App Demo

Release
v1

Report Template
Static Analysis Trace

BACK  GENERATE

<https://vulncat.fortify.com/en>



Fortify Taxonomy: Software Security Errors

This site presents a taxonomy of software security errors developed by the Fortify Software Security Research Group together with Dr. Gary McGraw. Each vulnerability category is accompanied by a detailed description of the issue with references to original sources, and code excerpts, where applicable, to better illustrate the problem.

The organization of the classification scheme is described with the help of terminology borrowed from Biology: vulnerability categories are referred to as phyla, while collections of vulnerability categories that share the same theme are referred to as kingdoms. Vulnerability phyla are classified into "seven plus one" pernicious kingdoms presented in the order of importance to software security:

1. [Input Validation and Representation](#)
2. [API Abuse](#)
3. [Security Features](#)
4. [Time and State](#)
5. [Errors](#)
6. [Code Quality](#)
7. [Encapsulation](#)
 - * [Environment](#)

The first seven kingdoms are associated with security defects in source code, while the last one describes security issues outside the actual code. To browse the kingdom and phylum descriptions, simply navigate the taxonomy tree on the left.

The primary goal of defining this taxonomy is to organize sets of security rules that can be used to help software developers understand the kinds of errors that have an impact on

Key FOD Tools and Plugins

Select the Tools from Main Dashboard.

1. Eclipse Security Assistant
2. FODUploader.jar

NAME	VERSION	RELEASE DATE	SUPPORTED	DOWNLOAD
Visual Studio Visual Studio Extension	Details	Details	Details	Up
Visual Studio VSTS / TFS Extension	Details	Details	Details	Up
Visual Studio Fortify Security Assistant for Visual Studio	Details	Details	Details	Up Down
eclipse Eclipse IDE Plugin	Details	Details	Details	Update site URL : https://tools.fortify.com/fodeclipseplugin
eclipse Fortify Security Assistant for Eclipse	Details	Details	Details	Up Down
IntelliJ IDEA IntelliJ Plugin	Details	Details	Details	Up
Jenkins Jenkins Plugin	Details	Details	Details	Up
MICRO FOCUS FOD Uploader JAVA	v3.1.0	2018/05/17	JRE 1.8 Installed on Build Server	Up Down

Key FOD Tools and Plugins -2

FOD Uploader

jFortify on Demand provides a build server integration (BSI) tool called FoDUploader to help you upload applications from a build server. FoDUploader runs from the command-line .

Usage:

```
java -jar FodUpload.jar -apiCredentials <apikey> -zipLocation appsrc.zip -entitlementPreference 2 -bsi <bsi Key>
```

Example:

```
java -jar FodUpload.jar -apiCredentials "ddda12dc-39e1-4ab9-91b0-94ac4fa6065f"  
"eE5TYjR6PHZLNzNzRUNxZ0hhV2o6dzlicVBsSFc20" -zipLocation /basic.mbs.zip -entitlementPreference 2 -bsi  
eyJ0ZW5hbnRJZCI6MTIxOSwidGVuYW50Q29kZSI6ImZvcnRpZnlfemwiLCJyZWxIYXNISWQiOjQyODUsInBheWxvYWRUe  
XBIIjoiQU5BTFTSVNfUEFZTE9BRCIsImFzc2Vzc21lbnRUeXBISWQiOjI1LCJ0ZWNo bm9sb2d5VHlwZSI6Ik1CUy9DL0MrKy9  
TY2FsYSIsInRIY2hub2xvZ3IUeXBISWQiOjE4LCJ0ZWNo bm9sb2d5VmVyc2lvbiI6bnVsbCwidGVjaG5vbG9neVZlcnPpb25JZ  
CI6bnVsbCwiYXVkaXRQcmVmZXJlbmNlIjoiQXV0b21hdGVkliwiYXVkaXRQcmVmZXJlbmNISWQiOjlsIm luY2x1ZGVUaGlyZ  
FBhcnR5IjpmYWxzZSwiaW5jbHVkZU9wZW5Tb3VyY2VBbmFseXNpcyl6ZmFsc2UsInBvcnRh bFVyaS I6Imh0dHBzO i8vc2F  
uZGJveC5mb3J0aWZ5LmNvbSIsImFwaVVyaS I6Imh0dHBzO i8vYXBpLnNhbmRib3guZm9ydGlmeS5jb20iLCJzY2FuUHJIz m  
VyZW5jZSI6IIN0YW5kYXJkliwic2NhblByZWZlcmVuY2VJZCI6MX0=
```

Authenticating

Beginning upload

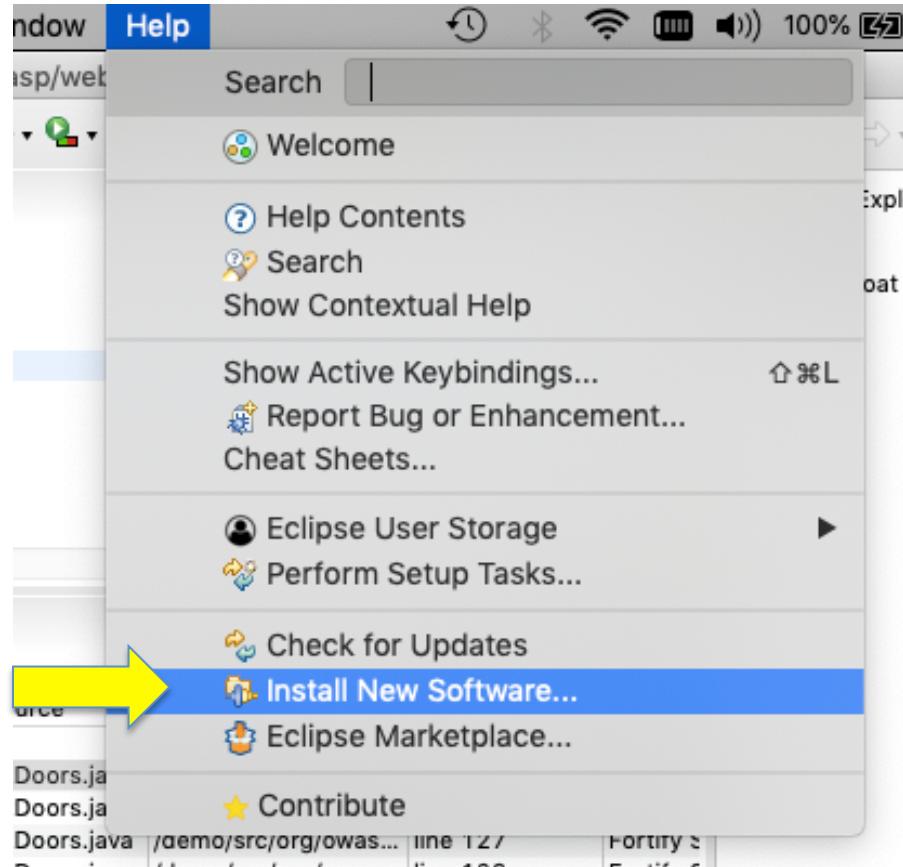
Upload Status - Bytes sent:98147

Scan 1871 uploaded successfully. Total bytes sent: 98147

Retiring Token : Token Retired Successfully

Key FOD Tools and Plugins – 3

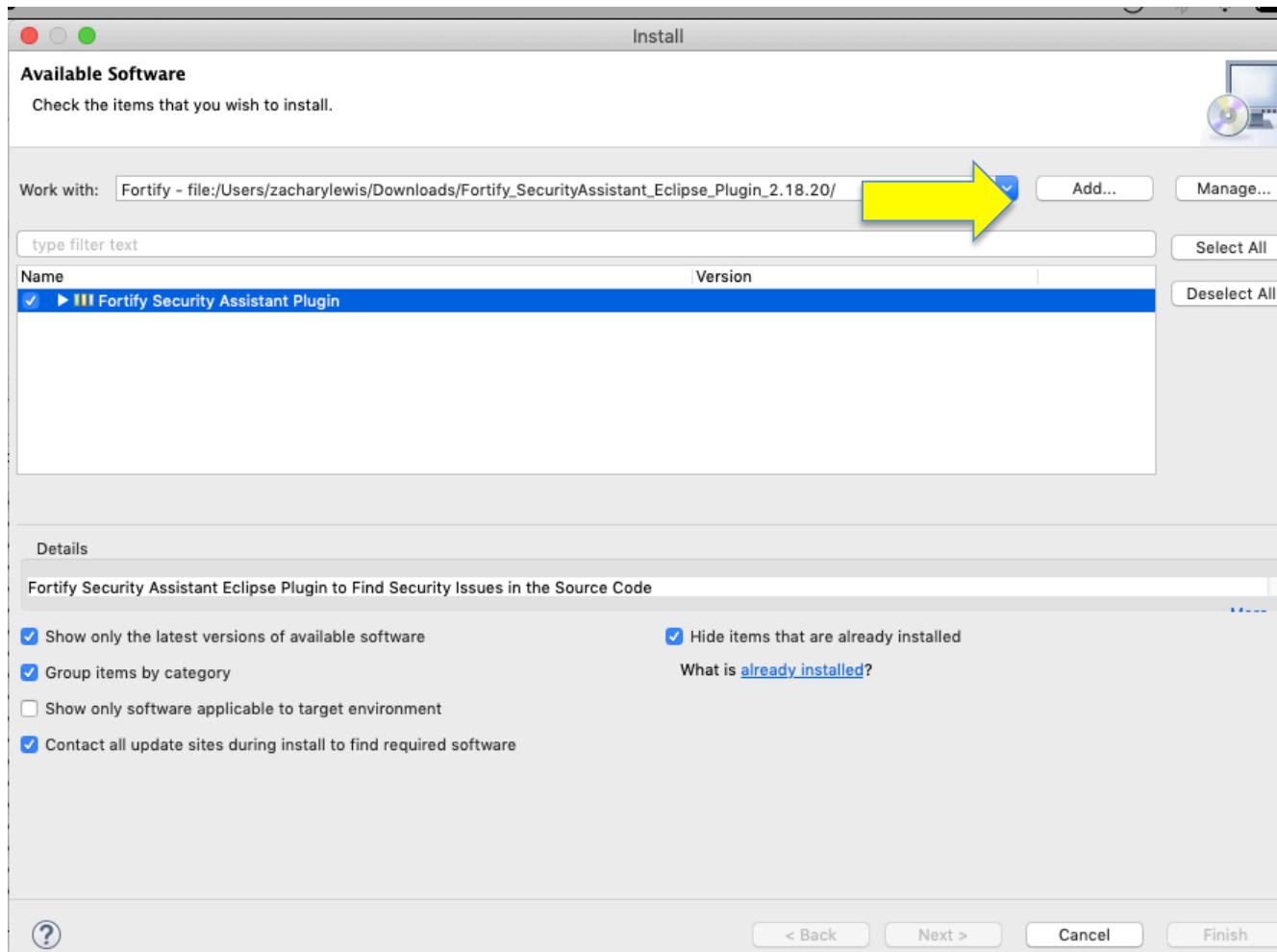
Installation of Eclipse Security Assistant Plugin



Key FOD Tools and Plugins – 4

Installation of Eclipse Security Assistant Plugin

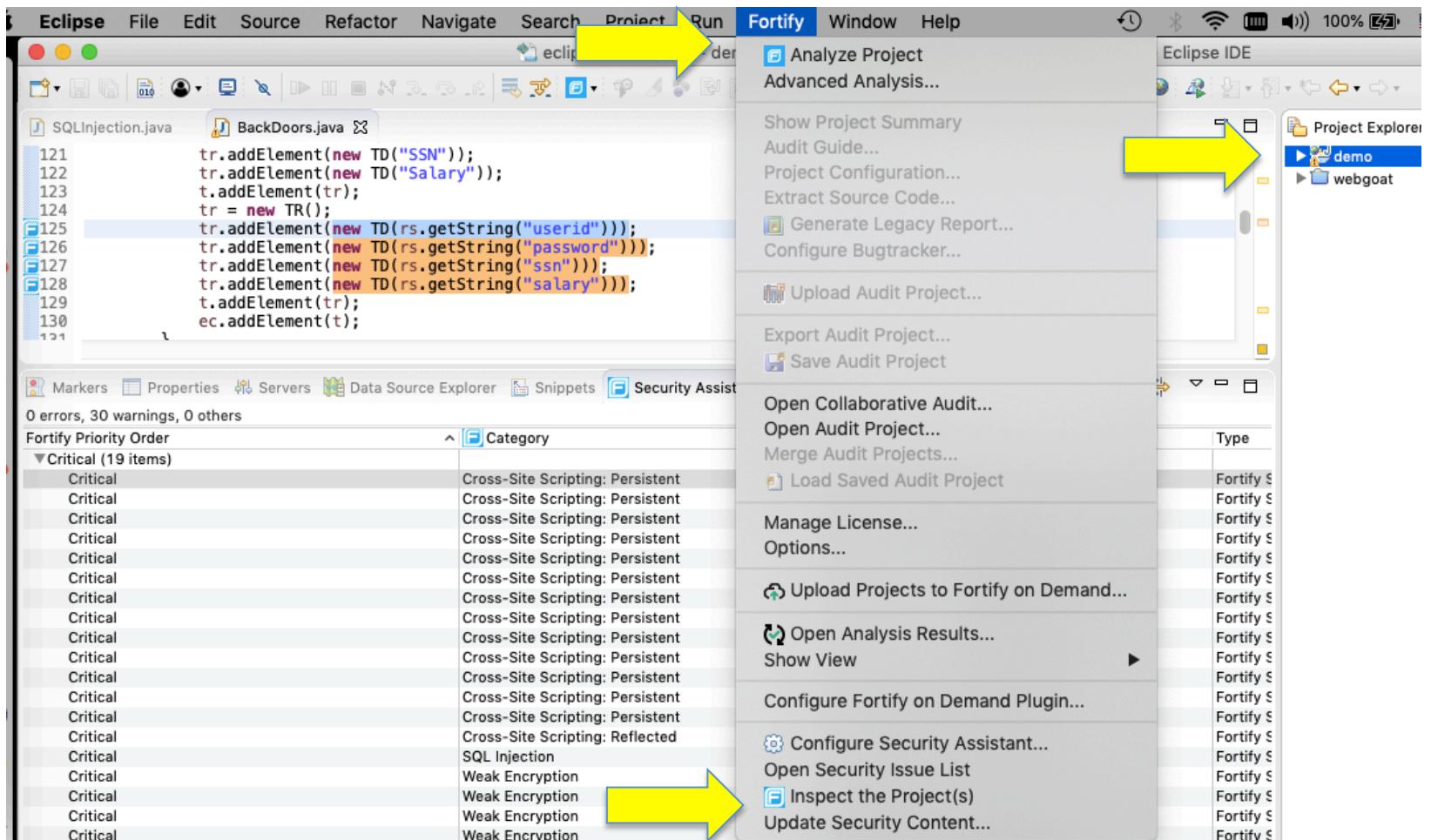
Add Path to Downloaded unzipped FOD Eclipse Security Assistant



Key FOD Tools and Plugins - 5

Using Eclipse Security Assistant Plugin

NOTE: It must be a Java Project and selected to use Inspect Project



Key FOD Tools and Plugins - 6

Using Eclipse Security Assistant Plugin

NOTE: It must be a Java Project and selected to use Inspect Project

The screenshot shows the Eclipse IDE interface with the following components visible:

- Project Explorer:** Shows the project structure with several Java projects and JRE System Library [JavaSE-1.8] containing various JAR files.
- Code Editor:** Displays the code for `BackDoors.java`. The code includes a loop that creates `TD` elements and adds them to a `tr` element, which is then added to a `tbody` element. A specific line of code is highlighted at line 125:

```
tr.addElement(new TD(rs.getString("userid")));
```
- Security Assistant Issues View:** This is the active view, showing the following details:
 - 0 errors, 32 warnings, 0 others
 - Fortify Priority Order
 - Category
 - A list of 20 critical issues, all categorized as "Cross-Site Scripting: Pers".
- Problems View:** Shows a list of 32 warnings, all categorized as "Cross-Site Scripting: Pers".
- Security Help View:** Provides information about Cross-Site Scripting, including an explanation, recommendation, and references.

Cross-Site Scripting: Persistent

[Explanation](#) [Recommendation](#) [References](#)

Explanation:

Cross-site scripting (XSS) vulnerabilities occur when:

1. Data enters a web application through an untrusted source. In the case of persistent (also known as stored) XSS, the untrusted source is typically a database or other back-end data store, while in the case of reflected XSS it is typically a web request. In this case, the data enters at `executeQuery` in `BackDoors.java` at line 113.
2. The data is included in dynamic content that is sent to a web user without being validated.

Key FOD Tools and Plugins - 7

Using Eclipse Security Assistant Plugin

NOTE: It must be a Java Project and selected to use Inspect Project

The screenshot shows a window titled "Security Help" with a close button. Below the title bar, there are navigation icons for back, forward, and other window operations. The main content area displays the following text:

Cross-Site Scripting: Persistent

[Explanation](#) [Recommendation](#) [References](#)

Explanation:

Cross-site scripting (XSS) vulnerabilities occur when:

1. Data enters a web application through an untrusted source. In the case of persistent (also known as stored) XSS, the untrusted source is typically a database or other back-end data store, while in the case of reflected XSS it is typically a web request.
In this case, the data enters at executeQuery in BackDoors.java at line 113.
2. The data is included in dynamic content that is sent to a web user without being validated.

Code Warrior AppSec Training

The screenshot shows the homepage of the Secure Code Warrior website. At the top, there's a navigation bar with links for REQUEST A DEMO, FAQS, SUBSCRIBE, and CONTACT US. Below the navigation is a main menu with links for ABOUT US, SOLUTION, PRICING, and RESOURCES. To the right of the menu are two buttons: a yellow 'PLAY NOW' button and a teal 'LOG-IN' button. The background features a dark, futuristic cityscape with glowing blue lights. Overlaid on this background are several pieces of text: 'SECURE CODE WARRIOR' in the top left, 'Gamified training. Real-time coaching.' and 'Anytime learning.' in the center, and 'SECURE YOUR CODE, FROM THE START.' in large orange letters at the bottom left. On the far right edge of the page, there's a vertical orange bar with the word 'LEVEL' written vertically on it.

SECURE CODE
WARRIOR

REQUEST A DEMO / FAQS / SUBSCRIBE / CONTACT US

ABOUT US SOLUTION PRICING RESOURCES

PLAY NOW LOG-IN

Gamified training. Real-time coaching.
Anytime learning.

SECURE YOUR CODE, FROM THE START.

LEVEL

Done