Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use, or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

© Copyright 2013 Hewlett-Packard Development Company, L.P.

Documentation Updates

The title page of this document contains the following identifying information:

• Software Version number
• Document Release Date, which changes each time the document is updated
• Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Part Number: 1-123-2013-09-400-01
## Contents

### Preface

- About Contacting HP Fortify Software ................................................................. vii
- Technical Support ................................................................................................... vii
- Corporate Headquarters ....................................................................................... vii
- Website ................................................................................................................ vii

- About the HP Fortify Software Security Center Documentation Set ...................... vii
- About HP Fortify Assistive Technologies ............................................................... viii

### Change Log ............................................................................................................ ix

### Chapter 1: Getting Started with HP Fortify Audit Workbench ................................. 10

- About Audit Workbench ......................................................................................... 10
- About Audit Workbench Projects and Project Templates ...................................... 10
  - About the Audit Guide Wizard ............................................................................ 10
  - About Hybrid 2.0 Technology ........................................................................... 10
  - About Customizable Reports ............................................................................. 10
  - About Integration with Software Security Center ............................................. 10
- About Upgrades ..................................................................................................... 11
  - Enabling HP Fortify Static Code Analyzer Suite Updates from HP Fortify Audit Workbench .......... 11
  - Upgrading Manually .......................................................................................... 13
  - Configuring Automatic Upgrades .................................................................. 13
- About Starting Audit Workbench .......................................................................... 14
  - Starting Audit Workbench on Windows Systems ............................................ 14
  - Starting Audit Workbench on Non-Windows Systems ..................................... 14
- About the Audit Workbench Launch Page ............................................................. 15
- About HP Fortify Security Content ...................................................................... 16
  - About Updating Security Content ................................................................... 16
  - Configuring Security Content Updates ............................................................ 17

### Chapter 2: Scanning Projects .................................................................................. 18

- Scanning a Java Project ........................................................................................ 18
- Scanning Complex Projects ................................................................................. 20
- About Quick Scan Mode ....................................................................................... 23
- Scanning HP Fortify Package for Microsoft Visual Studio Solutions .................... 24
- Re-scanning Projects ............................................................................................ 26

### Chapter 3: Viewing Scan Results in Audit Workbench ............................................ 27

- About the Issues Panel ......................................................................................... 28
- About Filters and Filter Sets ............................................................................... 28
- Creating Filter Sets .............................................................................................. 29
- Displaying Only Issues Assigned to You ............................................................. 31
- Viewing Suppressed, Removed, and Hidden Issues ............................................ 32
Chapter 4: Auditing Analysis Results

- Evaluating Issues
- Performing Quick Audits
- Creating Issues for Undetected Vulnerabilities
- Suppressing Issues

- Submitting Issues as Bugs
- About Hotspot Ranking of Unaudited Issues
- Listing Issues by Hotspot Ranking

- About Correlation Justification
- Using Correlation Justification

Chapter 5: Audit Workbench Reports

- About Default Report Templates
  - About the Fortify Security Report
  - About the Fortify Developer Workbook Report
  - About the OWASP Top Ten 2004, 2007, 2010 Reports
  - About the Fortify Scan Summary Report

- About Modifying Report Templates
- Viewing Report Templates
- Selecting Report Sections

- About Editing Report Subsections
- Saving Modified Report Template Settings
- About Report Template XML Files

- Generating Reports

Chapter 6: Functions View

- About the Functions View
  - Opening the Functions View
  - Sorting and Viewing Functions
  - Locating Functions in Source Code
  - Locating Classes in Source Code

- Determining Which Rules Matched a Function
- Writing Rules for Functions

- Creating Custom Cleanse Rules

Chapter 7: Troubleshooting

- Using the Debug Option
- Addressing the org.eclipse.swt.SWTErrror Error
- About Out of Memory Errors
  - Allocating More Memory for Static Code Analyzer

- Determining the Amount of Memory Used by External Processes
- Resetting the Default Views
Preface

About Contacting HP Fortify Software

If you have questions or comments about any part of this guide, contact HP Fortify using the information provided in the following sections.

Technical Support
650.735.2215
fortifytechsupport@hp.com

Corporate Headquarters
Moffett Towers
1140 Enterprise Way
Sunnyvale, CA 94089
650.358.5600
contact@fortify.com

Website
http://www.hpenterprisesecurity.com

About the HP Fortify Software Security Center Documentation Set

The HP Fortify Software Security Center documentation set contains installation, user, and deployment guides for all HP Fortify Software Security Center products and components. In addition, you will find technical notes and release notes that describe new features, known issues, and last-minute updates. The latest versions of these documents are available on the HP Software Product Manuals site:

http://h20230.www2.hp.com/selfsolve/manuals
About HP Fortify Assistive Technologies

In accordance with Section 508 of the U.S. Rehabilitation Act, HP Fortify Software Security Center, HP Fortify Audit Workbench, HP Fortify Plug-in for Eclipse, and HP Fortify for Package for Microsoft Visual Studio have been engineered to work with the JAWS screen reading software package from Freedom Scientific. JAWS provides text-to-speech support for use by the visually impaired. With JAWS, labels, text boxes, and other textual components can be read aloud, providing greater access to the information therein.

To generate text-to-speech translations in an HP Fortify product’s graphical user interface, you use standard JAWS commands. The following table lists keyboard combinations that can help you use JAWS with HP Fortify products. For more information on using JAWS, consult the JAWS documentation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Keyboard Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read values in combo boxes</td>
<td>Press <strong>CTRL + DOWN ARROW</strong> key or press <strong>ENTER</strong> to enable Form mode.</td>
</tr>
<tr>
<td>Tab through multi-line text boxes</td>
<td>Press <strong>CTRL + TAB</strong> to move from one multiline text box to the next.</td>
</tr>
<tr>
<td>Read multi-line labels</td>
<td>Press <strong>INSERT + DOWN ARROW</strong> to read all lines in label.</td>
</tr>
<tr>
<td>Read disabled items</td>
<td>Press <strong>INSERT + B</strong> or <strong>INSERT + DOWN ARROW</strong>.</td>
</tr>
<tr>
<td>Read disabled check boxes</td>
<td>Press <strong>ESC</strong> to switch Forms mode to Virtual Cursor mode.</td>
</tr>
</tbody>
</table>
| Enable reading of table headings | 1. Press **INSERT + F2**. The Run JAWS Manager dialog box opens.  
                                          2. Click **OK**. |
| Switch between pods or panels    | 1. Press and hold **CTRL + F7** as you select the new pane.  
                                          2. Release **CTRL + F7**. |
| Return focus to the application  | Press **CTRL + R** to refresh the display. If you do, your session ends and any data you have typed onto the page are lost. (JAWS reads the web browser application rather than the browser content.) |

For more information, visit the HP Accessibility & Aging website at **http://www.hp.com/accessibility**.

**Note:** For best results, run JAWS before you log on to HP Fortify Software Security Center.
## Change Log

The following table lists changes made to the *HP Fortify Audit Workbench User Guide*.

<table>
<thead>
<tr>
<th>Software Release-Version</th>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.00-01</td>
<td>06/04/13</td>
<td>Removed the section <em>About the Interface</em> from Chapter 1</td>
</tr>
<tr>
<td></td>
<td>06/11/13</td>
<td>Added the following sections to Chapter 1, <em>Getting Started with HP Fortify Audit Workbench</em>:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• About Upgrades</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>Enabling HP Fortify Static Code Analyzer Suite Updates from HP Fortify Audit Workbench</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Upgrading Manually</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>Configuring Automatic Upgrades</em></td>
</tr>
<tr>
<td></td>
<td>07/22/13</td>
<td>Added the section <em>Locating Classes in Source Code</em> to Chapter 6, <em>Functions View</em></td>
</tr>
</tbody>
</table>
Chapter 1: Getting Started with HP Fortify Audit Workbench

The following topics provide an overview of Audit Workbench, instructions on how to start the tool, and instructions on how to upgrade the Static Code Analyzer suite (SCA, Audit Workbench, and any plug-ins or packages you have installed) as new versions of the installer become available.

About Audit Workbench

Audit Workbench complements HP Fortify Static Code Analyzer (Static Code Analyzer) with a graphical user interface you can use to scan software projects and to organize, investigate, and prioritize the analysis results so that your team can fix security issues quickly and effectively.

From Audit Workbench, you can view and audit FPR files from HP Fortify Static Code Analyzer, HP Fortify Runtime Application Protection, and HP Fortify Program Trace Analyzer. Audit Workbench project templates help you sort the results of large scans in a way that works for your business and workflows.

About Audit Workbench Projects and Project Templates

In Software Security Center, a project is an application or code base that serves as a container for one or more project versions. A Software Security Center project version is an instance of the application or code base that is to be eventually deployed. An Audit Workbench project is comparable to a Software Security Center project version in that it represents a snapshot of the code base.

After you initiate a source code scan from Audit Workbench, Static Code Analyzer scans and analyses the code to produce comprehensive results. Audit Workbench organizes these results into a project.

Projects are defined by project templates, which determine how Audit Workbench (and Software Security Center) configure and prioritize the vulnerabilities (issues) uncovered in source code. Audit Workbench comes with a single basic project template, which you can use as is, or modify to suit your project needs. You can also import a project template from Software Security Center, or create a new project template from Audit Workbench.

About the Audit Guide Wizard

To get started with auditing, you can use Audit Workbench’s Audit Guide wizard. The Audit Guide helps you further filter issues and refine scan results to prepare for an audit.

About Hybrid 2.0 Technology

Audit Workbench’s Hybrid 2.0 technology connects penetration test results directly to source code analysis results to reveal hidden vulnerability relationships and expose their root causes within the source code. This enables your security and development teams to more accurately identify and prioritize vulnerabilities, and more productively investigate and remediate security defects in the source code.

About Customizable Reports

From Audit Workbench, you can generate customized reports based on any of the several baseline reports that come with the application.

About Integration with Software Security Center

Software Security Center provides a web portal that developers, managers, and security teams can use to share, collaborate, and track remediation of the potential vulnerabilities Static Code Analyzer scans uncover. If
you connect Audit Workbench to your Software Security Center instance, you can upload and merge your scan and audit results and share them with your team. This enables you to monitor trends and indicators across multiple project versions.

Integration with Software Security Center enables you to:

- Upload and download FPR files
- Manage the HP Fortify Secure Coding Rulepacks, custom Rulepacks, and external metadata applied during Static Code Analyzer scans
- Check for and install available upgrades of Static Code Analyzer and associated applications (including Audit Workbench)
- Download project templates
- Upload new and modified project templates
- Check permissions for custom tags

About Upgrades

You can check on the availability of new Static Code Analyzer suite (including Audit Workbench) versions from the Audit Workbench user interface. If a version newer than the one you have installed is available, you can download it and upgrade your instance.

You can also configure Audit Workbench to check for, download, and install new versions automatically at startup. Whether you upgrade your Static Code Analyzer and Apps manually or automatically, your data are preserved.

To upgrade Static Code Analyzer and Apps from Audit Workbench, a Software Security Center administrator must first set up the auto upgrade capability on the server host. The following topics address how to set up auto upgrades (as a Software Security Center administrator) for Audit Workbench and how to can perform the upgrades from Audit Workbench.

For information about the system requirements for using the auto upgrade feature, see the HP Fortify Software Security Center System Requirements document.

Enabling HP Fortify Static Code Analyzer Suite Updates from HP Fortify Audit Workbench

To make new Static Code Analyzer Suite installer available to Audit Workbench users for upgrades.

1. On the Software Security Center host, navigate to the <SSC_Install>/Core/support/tomcatForSSC/webapps/ssc/WEB-INF/internal directory and open the securityContext.xml file in a text editor.

2. Locate, and then uncomment the following line:

   <!-- <security:intercept-url pattern="/update-site/**" access="PERM_ANONYMOUS"/> -->

3. Save and close the securityContext.xml file.

4. Navigate to the <SSC_Install>/Core/support/tomcatForSSC/webapps/ssc/update-site/installers directory.

5. Open and read the readme.txt file.

6. Copy the sample update.xml file content (between and including the <installerInformation> and </installerInformation> tags, and then paste the copied sample text into a new text file.

7. Name the new file “update.xml” and save it to the <SSC_Install>/Core/support/tomcatForSSC/webapps/ssc/update-site/installers directory.

8. Any time a new Static Code Analyzer Suite installer file (HP_Fortify_SCA_and_Apps_<version>_<OS>.exe) becomes available, place it in the <SSC_Install>/Core/support/tomcatForSSC/webapps/ssc/update-site/installers directory.
9. Open the `update.xml` file in a text editor.
10. Between the `versionId` tags, type the version ID for the new installer.
    The version Id is the version number without the periods. For example, a version number of 4.00 corresponds to a version ID of 400.
    Make sure that the value of the “versionId” tag matches the Static Code Analyzer version in the installer.
Upgrading Manually

You can check for newer Static Code Analyzer and Apps versions manually, either from the Audit Workbench Help menu, or from the Options dialog box.

To check for, and (potentially) install, a newer Static Code Analyzer and Apps version, do one of the following:

- Select Help → Check for Upgrades.

Alternatively,

1. Select Options → Options.
   The Options dialog box opens.
2. In the left pane, leave Server Configuration selected.
3. In the Audit Workbench Upgrade Configuration section on the right, do the following:
   a. In the Server URL box, type the URL for your Software Security Center server.
   b. Click Check Now.

The upgrade tool polls the upgrade server for information about the Static Code Analyzer and Apps versions available for the platform on which it is running. If a newer version is available, the upgrade tool prompts you to indicate whether you want to proceed to download and install it.

Configuring Automatic Upgrades

To configure upgrade checks at startup:

1. Select Options → Options.
   The Options dialog box opens.
2. In the left pane, leave Server Configuration selected.
3. In the Audit Workbench Upgrade Configuration section on the right, do the following:
   a. In the Server URL box, type the URL for the installers folder on your Software Security Center server.
   b. Select the Check for upgrades at startup check box.
4. Click OK.

After this, each time you start Audit Workbench, the auto upgrade tool checks the server to determine whether a newer SCA and Apps version is available and then, if a newer version is available, downloads and installs it.
About Starting Audit Workbench

You can start Audit Workbench from the start menu on a Windows system. You can start it from the command line on any supported operating system.

Starting Audit Workbench on Windows Systems

To start Audit Workbench on a Windows system, do one of the following:

- Select Start → All Programs → HP Fortify SCA and Applications 4.00 → Audit Workbench.

Alternatively,

1. Open a command prompt window, and then change to the C:\Program Files\HP_Fortify\HP_Fortify_SCA_and_Apps_4.00\bin directory.
2. At the prompt, type auditworkbench.cmd.

Starting Audit Workbench on Non-Windows Systems

To start Audit Workbench on non-Windows systems:

1. Open a command prompt window, and then change to the C:\Program Files\HP_Fortify\HP_Fortify_SCA_and_Apps_4.00\bin directory.
2. At the prompt, type auditworkbench.
About the Audit Workbench Launch Page

The launch page contains the following sections:

- The **Start New Project** section provides links that you can use to start source code scanning wizards.
- The **Open Project** section displays links to any projects recently opened in Audit Workbench. To open a project, click its name, or use the **Open Project** link to browse to the project.
- After you provide your Software Security Center credentials (through the **Sign In** link), the **Open Collaborative Audit** section lists the server projects and versions to which you have access.

**Note:** HP Fortify Audit Workbench comes with several code samples to use to help you learn to use the tool. For information about these samples, see *Sample Files* on page 94.
About HP Fortify Security Content

HP Fortify security content (security content) consists of Secure Coding Rulepacks and external metadata. The external metadata include mappings from the HP Fortify categories to alternative categories (such as OWASP 2010, PCI 1.2, and CWE). You can modify the existing mapping in the external metadata document (externalmetadata.xml) or create your own files to map HP Fortify issues to different taxonomies, such as internal application security standards or additional compliance obligations (recommended).

Use any XML editor to make your changes or create a new document. (The existing mapping file is located in the `<SCA_and_Apps_Install>\Core\config\ExternalMetadata` directory.) HP Fortify recommends that you save your new or modified document to the `<SCA_and_Apps_Install>\Core\config\CustomExternalMetadata` directory so that your changes are not lost during security content updates.

To validate a modified or new mapping, use the `externalmetadata.xsd` file, which is located in the `<SCA_and_Apps_Install>\Core\config\schemas` directory. HP recommends that, after you change your mapping document, you open the FPR file in the plug-in to see how the mapping works with the scan results.

If you change the external metadata document or create a new mapping document, be sure to make the same changes on Software Security Center.

**Note:** When you update security content, any changes made locally to the Secure Coding Rulepacks and external metadata are overwritten.

About Updating Security Content

You can obtain the latest security content by doing one of the following:

- Configure the security content update server.
- Poll the security content update server for scheduled updates
- Manually copy any custom rules to the `<SCA_and_Apps_Install>\Core\config\customrules` directory.
- Manually copy any custom external metadata files to the `<SCA_and_Apps_Install>\Core\config\CustomExternalMetadata` directory.

The following topics provide information about how to update HP Fortify Security Content (Security Content) and manage security settings.
Configuring Security Content Updates

To configure the security content updates:

1. Select **Options → Options**.
   
The Options dialog box opens to the **Server Configuration** section.

2. To update security content from your Software Security Center instance, in the **Security Content Update Configuration** section, select the **Update Security Content from Software Security Center** check box.

3. To specify an update server from which to update security content, in the **Security Content Update Configuration** section, do the following:
   
a. In the **Server URL** box, type the URL for the update server.

b. If required, in the **Proxy Server** and **Port** boxes, type the proxy server and port, respectively.

4. To configure automatic updates, select the **Perform Security Content Update Automatically** check box, and then in the **Security Content Update Frequency (Days)** box, specify how often (type the number of days) the security content is to be automatically updated.

5. Click **OK**.
Chapter 2: Scanning Projects

The following topics describe how to scan source code and view the scan and analysis results in the Audit Workbench auditing interface.

Scanning a Java Project

The Scan Java Project wizard combines the translation and analysis phases of the scanning process into a simple step. Use this wizard to scan small Java projects that have source code in a single directory.

To scan a new Java project:

1. Open Audit Workbench.
2. In the **Start New Project** section, click **Scan Java Project**.
   - The Browse for Folder dialog box opens.
3. Select the folder that contains all the source code you want to analyze, and then click **OK**.
   - **Note:** Static Code Analyzer sets the build ID to the folder name.
   - The Java Version dialog box opens.
4. Select the Java version used for your project, and then click **OK**.
   - The Audit Guide Wizard opens.
5. Select the settings for the types of issues you want to display in the results, and then click **Run Scan**.
   - Static Code Analyzer analyzes the source code. If Static Code Analyzer encounters any problems as it scans the source code, it displays a Warning dialog box.
6. Click **OK**.

After the scan is completed, Audit Workbench displays the analysis results.
**Note:** Static Code Analyzer scans invoked from Audit Workbench are invoked with the server Java Virtual Machine.
Scanning Complex Projects

Exceptionally large code bases may require distinct measures to ensure a complete scan, including using Static Code Analyzer to scan the code in smaller sections. While Audit Workbench allows editing of SCA command parameters, large, complex scans are more successfully handled directly through the command console. In addition, if a system has memory constraints, SCA must compete with the Fortify Audit Workbench for resources, possibly resulting in slow or failed scans.

Use the Advanced Scan wizard to translate and analyze JavaScript, PHP, ASP, .NET, and SQL projects. You can use the wizard for Java projects that have source code in multiple directories, special translation or build conditions, or that have files that you want to exclude from the project.

**Note:** Audit Workbench automatically filters out unsupported files within the selected source code directories.

To scan a new project:

1. Open Audit Workbench.
   The launch page displays.

2. In the **Start New Project** section, click **Advanced Scan**.
   The Browse for Folder dialog box opens.

3. Select the root directory of the project, and then click **OK**.
   The Commandline Builder opens.

   The wizard automatically includes all supported files in the scan.

4. (Optional) To add files from another directory:
   a. Click **Add Directory**.
      The Browse to Folder dialog box opens.
   b. Select the folder that contains the files you want to add to the scan, and then click **OK**.
The navigation panel displays the directory and Audit Workbench adds all supported files to the scan. (To remove the directory, right-click the folder, and then select **Remove Root** from the shortcut menu.)

5. (Optional) To exclude files or directories that contain, for example, test source code, right-click the file or directory, and then select **Exclude** from the shortcut menu.

6. For Java projects, set the following:
   a. Select the build directories and jar files and click **Classpath Directory**.
      
      **Note**: If you do not select the classpath directory, the CLASSPATH environment variable is used. The folder turns blue and the files are added to the classpath.
   b. From the **Specify Java Version** list, select the Java version of the project.

7. In the **Specify Build ID** box, type the build ID. (The root directory is the default build ID.)

8. To specify a different output file path than the default, in the **Specify Build ID** box, type the path and file name for the FPR file that Static Code Analyzer is to generate.

9. To perform a quick scan, check the **Enable Quick Scan Mode** check box. For information about quick scans, see **About Quick Scan Mode** on page 23.

10. Click **Next**.
    
    The Commandline Builder opens.

    ![Commandline Builder](image)

The Static Code Analyzer scan process includes the following stages:

- During the **clean** phase, Static Code Analyzer removes files from previous translation of the project.
- During the **translation** stage, Static Code Analyzer translates source code identified in the previous screen into an intermediate format that is associated with a build ID. The build ID is typically the project.
- During the **scan** stage, Static Code Analyzer scans source files identified during the translation phase and generates analysis results, in the HP Fortify Project (FPR) format.
11. (Optional) To skip a scanning phase, clear the **Enable Clean**, **Enable Translation**, or **Enable Scan** check box.
   For example, if the security content has changed but the project has not changed, you might want to disable the clean stage so that Static Code Analyzer scans the project without retranslating.

12. Modify the command-line options for each Static Code Analyzer scan stage, as required.

13. (Optional) To analyze the source code using a custom Rulepack, or to disable a Rulepack, do the following:
   a. Click **Configure Rulepacks**.
      
      The Additional Options dialog box opens.

      ![Add Custom Rulepack](image)

      b. In the **Fortify Secure Coding Rulepacks** list, clear the check boxes that correspond to any Rulepacks you want to disable during the scan.
   c. To add a custom Rulepack, click **Add Custom Rulepack**, and then browse to and select the Rulepack file.
   d. Click **OK**.
14. From the Commandline Builder, click **Next**.

15. Select your scan settings, and then click **Run Scan**.

Static Code Analyzer starts the scan and displays progress information throughout the process. If Static Code Analyzer encounters any problems scanning the source code, it displays a warning.

After the scan is completed, Audit Workbench loads the audit project and displays the analysis results.

**About Quick Scan Mode**

You can use Quick Scan Mode to quickly scan your projects for high-confidence, high-severity issues. Keep in mind that, although Quick Scan Mode is significantly faster than a full scan, it does not provide a robust results set. A quick scan of the webgoat sample project detects 120 issues. By contrast, a full scan of the WebGoat sample detects 1,549 issues.

With Quick Scan Mode enabled, Static Code Analyzer scans your project using the **fortify-sca-quickscan.properties** file, rather than the standard **fortify-sca.properties** file used for full scans. You can edit the **fortify-sca-quickscan.properties** file to specify other parameters to search for.

To perform a quick scan, follow the steps described in *Scanning Complex Projects* on page 20 and select the **Enable Quick Scan Mode** check box. Audit Workbench displays the scan results in its **Project Summary** panel. You audit quick scan results just as you audit full scan results.
Scanning HP Fortify Package for Microsoft Visual Studio Solutions

You can analyze HP Fortify Visual Studio Package solutions and projects. To scan HP Fortify Visual Studio Package projects or solutions, you must have Visual Studio and the HP Fortify Package for Microsoft Visual Studio installed on the same machine as Audit Workbench.

The source code analysis is powered by Static Code Analyzer and HP Fortify Secure Coding Rulepacks, and supports the following languages in Visual Studio solutions:

- C/C++
- C#
- Visual Basic .NET
- ASP .NET

To scan a Visual Studio solution:

1. Start Audit Workbench.
2. In the Start New Project section, click Visual Studio Build Integration. The Browse for Folder dialog box opens.
3. Select the folder that contains the solution you want to analyze, and then click OK.

   **Note:** Static Code Analyzer uses the selected folder name as the build ID.

   The Commandline Builder opens.

4. Configure the solution settings, as follows:
   a. Next to the Specify Visual Studio Solution File box, click Browse.
   b. Navigate to and select the file for your Visual Studio solution.
c. From the Specify Visual Studio Version list, select the Visual Studio version used for the solution.
d. In the Specify Build Configuration box, leave the default value (DEBUG).
e. (Optional) In the Specify Build ID box, type a different build ID.
f. (Optional) Specify a different path and name for the results file.
g. Click Next.


The Commandline Builder displays details about the Static Code Analyzer analysis stages for the scan.

• During the clean phase, Static Code Analyzer removes files from previous translation of the project.
• During the translation stage, Static Code Analyzer translates source code identified in the previous screen into an intermediate format that is associated with a build ID. The build ID is typically the project.
• During the scan stage, Static Code Analyzer scans source files identified during the translation phase and generates analysis results, in the HP Fortify Project (FPR) format.

6. (Optional) To skip a scanning phase, clear the Enable Clean, Enable Translation, or Enable Scan check box. For example, if the Rulepacks have changed but the project has not changed, you might want to disable the clean stage so that Static Code Analyzer scans project without retranslating.

7. Modify the command-line options for each Static Code Analyzer phase, if necessary.

8. (Optional) To analyze the source code using a custom Rulepack, or to disable a Rulepack, do the following:
   a. Click Configure Rulepacks.
      The Additional Options dialog box opens.
   b. In the Fortify Secure Coding Rulepacks list, clear the check boxes that correspond to any Rulepacks you want to disable during the scan.
   c. To add a custom Rulepack, click Add Custom Rulepack, and then browse to and select the Rulepack file.
   d. Click OK.

9. From the Commandline Builder, click Next.

10. Select your scan settings, and then click Run Scan.

Static Code Analyzer starts the scan and displays progress information throughout the process. If Static Code Analyzer encounters any problems scanning the source code, it displays a warning.

After the scan is completed, Audit Workbench loads the audit project and displays the analysis results.
Re-scanning Projects

This section explains how to re-scan a project that was translated locally with new or updated rules. Audit Workbench automatically loads the FPR project settings such as the build ID and source code path, and allows you to change the command-line scanning options.

After Static Code Analyzer completes the scan, Audit Workbench merges the analysis results with those from the previous scan to determine which issues are new, which have been removed, and which were uncovered in both scans.

To re-scan a project:
1. Open an FPR file.
2. Click Scan. The Rescan Build ID dialog box opens. If you do not have a Build ID, Audit Workbench displays a warning.
3. If the source code has changed since the most recent scan, click Update Project Translation to retranslate the project.
   Note: If the FPR file that you opened was generated by a Static Code Analyzer scan that was not initiated from Audit Workbench, the Update Project Translation button is greyed out.
   Note: If the source code has changed since the most recent scan, you must update the translation before you re-scan the code. Otherwise, a new scan cannot uncover the issues in the updated source code.
4. (Optional) Modify the Static Code Analyzer scan phase command-line options, as necessary.
5. (Optional) To change the Rulepacks used to analyze the project:
   a. Click Configure Rulepacks.
   b. To add and remove Rulepacks, select or clear the check boxes, as necessary. (To use a custom Rulepack that is not listed, click Add Custom Rulepack, and then browse to and select the Rulepack file.)
   c. Click OK.
6. Click Run Scan. After the scan is completed, Audit Workbench displays the results.
7. Compare the new results with the issues uncovered in the previous scan as follows:
   - To display all new issues, click the All Folder tab, and then select the Group by New Issue option.
   - To display removed issues, click the All Folder tab, and then select Options → Show Removed Items.
   - To review issues found in both the previous scan and the new scan, click the All Folder tab, expand the Issue Updated group, and then, from the Group by list, select New Issue.
Chapter 3: Viewing Scan Results in Audit Workbench

After a scan is completed, Audit Workbench displays the results in the auditing interface. The following screen capture shows the WebGoat sample FPR file in the auditing interface.

The auditing interface consists of the following sections, which are numbered in the screen capture:

1. Issues panel
2. Source code panel
3. Functions panel
4. Issue auditing panel
5. Analysis Evidence panel

These sections are described in the following topics.
Note: The graphical user interface of Audit Workbench responds to system font size changes. If you enlarge the system font, Audit Workbench enlarges all of its text elements accordingly, including the text displayed in the source code editor.

About the Issues Panel

The issues panel in the upper left portion of the auditing interface provides a way to group and select issues for auditing. The panel contains the Filter Set list, folders (tabs), the Group By list, the My Issues check box, and a search box.

About Filters and Filter Sets

Audit Workbench applies filters to sort and filter the issues that Static Code Analyzer uncovers. Audit Workbench organizes filters into distinct filter sets.

Each project can have unique filter sets because the filter sets are saved in a project file. A project template can contain definitions for multiple filter sets. Using multiple filter sets in a project enables you to quickly change the sort order and visibility of issues.

The filter set you select from the Filter Set list determines which issues are displayed in the auditing interface. The filter set customizes the analysis results panel by determining the number and types of containers (folders) and how and Audit Workbench displays issues.

The default filter sets include: Security Auditor View, Developer View, Critical Exposure, and Hotspot. The filter sets sort the issues by priority into the Critical, High, Medium, and Low folders. All default filter sets have the same sorting mechanism.

To change the issues displayed in the auditing interface, from the Filter Set list, select one of the following:

- **Security Auditor View**: This view uses the most comprehensive set of rules to reveal a broad set of security issues to be audited. The Security Auditor View filter contains no visibility filters, so all issues are shown.
- **Developer View**: Issues shown include a balance between results that detail all potential issues and a targeted set of possible vulnerabilities.
- **Critical Exposure**: Shows issues within categories that have been proven to be high priority issues across multiple industries and within a variety of environments; used to discover a limited set of well-known, critical security issues.
- **Hotspot**: Shows issues that are of particular interest to developers, such as high accuracy bugs. Issues are sorted into one of the following four folders based on the Hotspot calculations:
  - Issues that have been audited, and runtime issues, are the basis of the HotSpot calculation, and are all placed into the Evidence folder.
  - If Hotspot scores indicate that issues are likely to be correctly audited problems, issues are placed into the Likely and Possible folders.
  - If Hotspot scores indicate that issues are not likely to be audited as problems, issues are placed in the Unlikely folder.
  - Finally, issues with a Hotspot score that provides no meaningful information are placed in the Unknown folder.
- **Data Validation**: Sorts issues into six folders based on the type of data validation used. Looks for HTML encoding, URL encoding, other types of encoding, Struts Validation, and .NET Validation. HP Fortify recommends that you use the Priority by Category grouping with this setting.

Note: The filter set value is set to Developer View by default. This may result in some issues not being visible. Decrease the filter set value to Security Auditor View to ease the filter, and to display more issues.
Creating Filter Sets

If the filter sets available in Audit Workbench do not exactly suit your needs, you can create your own, either by using the filter wizard, or by copying and then modifying an existing filter set.

Creating New Filter Sets Based on Existing Filter Sets

To copy an existing filter set:

1. Select **Tools → Project Configuration**
2. In the left pane of the **Filter Sets** tab, click the plus character (+).
   The Create New Filter Set dialog box opens.
3. In the text box, type a name for the filter set.
4. Select a filter set to copy.
5. Click **OK**

Audit Workbench lists the new (copied) filter set on the **Filter Sets** tab of the Project Configuration dialog box, and adds it to the **Filter Set** list in the issues view.

Creating Filters from the Issues Panel

If you find an issue in a folder list that you want to hide or direct to another folder, you can create a new filter using the filter wizard. The filter wizard displays all the attributes with matching conditions for the filter.

To create a new filter from an issue:

1. From the **Filter Sets** list, select a filter set.
2. In analysis results panel, select an issue.
3. Right-click and select **Generate Filter**.
   The Create Filter dialog displays a list of suggested conditions.
4. (Optional) To expand the conditions list, select **Show all conditions**.
   The Create Filter dialog displays additional conditions.
5. Select the conditions to use in the filter. You can fine tune the filter later from the **Filter** tab.
6. Select the type of filter to create, as follows:
   - Select **Hide Issue** to create a visibility filter.
   - Select **Set Folder** to create a folder filter, and then select the folder name or select Other Folder to add an existing folder or create a new one.
     A new folder appears only in this filter set.
7. Click **Create Filter**.
   The wizard places the new filter at the end of the filter list. For folder filters, this gives the new filter the highest priority. Issues matching the new folder filter appear in the targeted folder.
8. (Optional) For folder filters, drag the filter higher in the folder filter list to change the priority.

Audit Workbench sorts the issues using the new filter.

**Note:** The filter is created only in the selected filter set.
Creating Visibility Filters on the Filters Tab

From the Filters tab, you can create visibility filters for attributes and values. The filter you create applies only to the selected filter set.

To create a new visibility filter on the Filters tab:
1. From the Filter Set list in the issues panel (upper left), select a filter set to which you want to apply the new filter.
2. Select Options → Show View → Filters.
   Audit Workbench displays the Filters tab to the right of the Analysis Evidence panel (center bottom).
3. On the Filters tab, right-click Visibility Filters, and then select Create New Filter on the shortcut menu.
   Audit Workbench displays the Create Filter dialog box.
4. From the first (left-most) list in the If panel, select an issue attribute.
5. From the second list in the If panel, select an operator for the filter.
6. From the third (right-most) list, select an attribute value (or range of values).
7. In the lower Then panel, leave Hide Issue selected.
8. Click Save.
Audit Workbench places your new filter at the end of the Visibility Filters list and thereafter uses the new filter in sorting issues.

Note: Audit Workbench creates the filter only in the filter set you selected.

Creating Folder Filters on the Filters Tab

From the Filters tab, you can create folder filters for attributes and values to apply to a specific filter set.

To create a new filter on the Filters tab:
1. From the Filter Set list in the issues panel (upper left), select a filter set to which you want to apply the new filter.
2. Select Options → Show View → Filters.
   Audit Workbench displays the Filters tab to the right of the Analysis Evidence panel (center bottom).
3. On the Filters tab, right-click Folder Filters, and then select Create New Filter on the shortcut menu.
   Audit Workbench displays the Create Filter dialog box.
4. From the first (left-most) list in the If panel, select an issue attribute.
5. From the second list in the If panel, select an operator for the filter.
6. From the third (right-most) list, select an attribute value (or range of values).
7. In the lower Then panel, from the Set to Folder to list, select one of the following:
   • To direct issues that match the new folder filter to an existing folder, select the folder name.
   • To direct issues that match the new folder filter to a new folder, select Other Folder, and then create a new folder.
   • To direct issues that match the new folder filter to a folder that you have created, select Other Folder, and then select the existing folder.
   Audit Workbench displays your new filter at the end of the Folder Filters list. This gives the new filter the highest priority. Audit Workbench applies folder filters in the order listed on the Filters tab and directs issues to the last folder filter they match in the list.
8. (Optional) To change the priority of a filter, drag it higher in the Folder Filters list.

Note: Audit Workbench creates the filter only in the filter set you selected.
Copying Filters from One Filter Set to Another

Although filter settings are local to the filter set, you can copy a filter to another filter set in the project. If you copy a folder filter to a different filter set, and that folder is not already active in the set, Audit Workbench adds the folder automatically.

To copy a filter:
1. From the Filter Set list, select a filter set.
2. On the Filters tab, select a filter.
3. Right-click, and select Copy Filter to.
   - The Select a Filter Set dialog box lists of all filter sets.
4. Select a filter set, and then click OK.
   - The filter is added to the filter set in the last position.
5. (Optional for folder filters) To change the order of a folder filter in the list, drag it to a different position in the list.
   - Audit Workbench re-sorts the issues based on the new filters.

Adding Folders to Filter Sets

Use this section to enable an existing folder in a filter set. Create a new folder that only appears in the selected filter set using the instructions Creating Folders on page 33. To display issues in this folder, create a filter that targets the new folder.

To add a folder:
1. Select Tools → Project Configuration.
   - Audit Workbench displays the Project Configuration dialog box.
2. Click the Folders tab.
3. From the Folder for Filter Set list, select the filter set in which you want the folder displayed.
   - The Folder for Filter Set list filters the folders displayed in the folder list. If you select All Folders, all folders that are defined in the project template display in the list.
4. Click + (plus symbol) next to Folders.
   - The Add new folder to the Filter Set dialog box opens. If all folders already display in this filter set, the Create New Folder dialog box opens.
5. Select the folder you want to add, and then click Select.
   - The folder appears in the folder list.
6. Click OK.
   - Audit Workbench displays the folder as a tab along with the other folders.

Displaying Only Issues Assigned to You

To have the issues panel display only issues that have been assigned to you, do one of the following:

- Select the My Issues check box.
- Select Options → Show Only My Issues.
Viewing Suppressed, Removed, and Hidden Issues

You can control whether the issues panel lists the following types of issues:

- **Suppressed issues.** As you assess successive scans of a project version, you might want to completely suppress some exposed issues. It is useful to mark an issue as suppressed if you are sure that the specific vulnerability is not, and will never be, an issue of concern. You might also want to suppress warnings for specific types of issues that might not be high priority or of immediate concern. For example, you can suppress issues that are fixed, or issues that you plan not to fix.

- **Removed issues.** As multiple scans are run on a project over time, issues are often remediated or become obsolete. Static Code Analysis marks issues that were uncovered in a previous scan, but are no longer evident in the most recent SCA analysis results as Removed Issues.

- **Hidden issues.** Issues are hidden if Audit Workbench visibility filters such as the Developer View and Critical Exposure filter sets have excluded them from display.

To hide or show suppressed, removed, or hidden issues in the issues panel:

- Select **Options**, and then select (or deselect) one or more of the following:
  - Show Suppressed Issues
  - Show Removed Issues
  - Show Hidden Issues

About Folders

Audit Workbench organizes scan results for a project into folders, which contain logically defined sets of issues. Audit Workbench displays the folders as color-coded tabs in the issues panel (top left panel). The number of issues that each folder contains is displayed at the top of each tab.

Audit Workbench comes with five folders. The filter set you select (**Filter Set** list) determines which folders are visible in the issues panel. The following folders are visible while the default Security Auditor View filter set is selected:

- The **Critical** folder contains issues that have a high impact and a high likelihood of occurring. Issues at this risk level are easy to discover and to exploit, and represent the highest security risk to a program. Remediate critical issues immediately.
Example: SQL Injection

- The High folder contains issues that have a high impact and a low likelihood of occurring. High-priority issues are often difficult to discover and exploit, but can result in much asset damage. They represent a significant security risk to a program. Remedy these issues with the next patch release.

Example: Password Management: Hardcoded Password

- The Medium folder contains issues that have a low impact and a high likelihood of exploitation. Medium-priority issues are easy to discover and exploit, but often result in little asset damage. These issues represent a moderate security risk to a program. Remedy these issues as time permits.

Example: ASP.NET Misconfiguration: Missing Error Handling

- The Low folder contains issues that have a low impact and a low likelihood of exploitation. Remedy these issues as time permits. Low-priority issues can be difficult to discover and to exploit and typically result in little asset damage. These issues represent a minor security risk to the program.

Poor Error Handling: Empty Catch Block

- The All folder contains the issues from all the other folders.

An issue is listed in a folder if the folder filter conditions match the issue attributes. One folder in each filter set is the default folder, indicated by (default) in the folder name. If an issue does not match any of the folder filters, the issue is listed in the default folder.

You can create your own folders as you need them. For example, you might group all hot issues for a project into a Hot folder and group all warning issues for the same project into a Warning folder. For instructions on how to create your own folders, see Creating Folders.

Creating Folders

You can create your own folders as you need them. To display issues in a new folder, create a folder filter that targets the new folder.

To create a new folder:

1. Select Tools → Project Configuration.
   The Project Configuration dialog box opens.
2. Click the Folders tab.
   The panel on the left displays existing folders. Fields on the right show the filter set, name, color, and description of the selected folder.
3. Select a filter set to enable a folder that displays in the selected filter set only from the Folder for Filter Set list. The Folders for Filter Set filters the folders displayed in the folder list.
4. To add a folder:
   a. Click the plus icon next to Folders.
      The Create New Folder dialog box opens.
   b. Type a unique name for the folder, select a folder color, and then click OK.
      Audit Workbench displays the folder name at the bottom of the folder list.
5. In the Description box, type a description of the issues the folder is designed to list.
6. (Optional) To change the tab position in the issues panel, drag the folder up or down.

7. (Optional) To place all issues that do not match an existing folder filter into the new folder, select the Default Folder check box.

8. Click OK.

Audit Workbench displays the folder as a tab along with the other folders. For instructions on how to add folders to a filter set, see Adding Folders to Filter Sets on page 31.

**Renaming Folders**

If you rename a folder, the name change is global and is reflected in all filter sets.

To rename a folder:

1. Select Tools → Project Configuration.
   The Project Configuration dialog box opens.
2. Click the Folders tab.
3. From the Folders for Filter Set list, select a filter set that displays the folder you want to rename.
4. In the Folders panel, select the folder to rename.
5. In the Name box, select the existing folder name, and then type a new name.
   In the Folders panel, the folder name changes as you type.
6. Click OK.

The new folder name displays on the tabs.

**Removing a Folder from a Specific Filter Set**

The following procedure describes how to remove a folder from a filter set without removing it from other filter sets.

To remove a folder from a specific filter set:

1. Select Tools → Project Configuration.
   The Project Configuration dialog box opens.
2. Click the Folders tab.
3. From the Folders for Filter Set list, select a filter set.
4. Audit Workbench removes the folder from the selected filter set only.
   The folder list displays the folders in the selected filter set.
   If the folder is a target of a folder filter, Audit Workbench hides the option to remove the folder.
5. Select the folder and click the minus character (-) next to Folders.
   If the folder is a target of a folder filter, the Conflicts Occurred Removing a Folder dialog box opens.
6. Retarget or delete Folder Filters, as required.
   The folder list no longer includes from the removed folder.
7. Click OK.

Audit Workbench no longer displays the folder as a tab.
About Grouping Issues

The items visible in the navigation tree vary based on the Group By option selected in the analysis results panel. The value you select from the Group By list sorts issues in all visible folders into subfolders.

To list all issues in a folder without any grouping, select <none>.

You can customize the existing groups by changing which attributes the groups are sorted by, by adding or removing attributes to create sub-groupings, or by adding your own group options.

The Group By list options are for the application instance. You can apply a grouping option to any project opened with that instance of the application. You can view issues using any of the grouping options, and you can create and edit groups.

Creating Grouping Options

You can create a grouping option that groups issues in a hierarchical format in sequential order based on specific attributes.

To create a new grouping option:

1. From the Group By list, select Edit.
   The Edit Custom Groupings dialog box opens.
2. From the Custom Group Name list, select Create New.
   Audit Workbench displays the Enter Value dialog box.
3. In the text box, type a name for the new custom group.
4. Click OK.
5. From the Grouping Types list on the left select a type, and then click the right-pointing arrow to move the option to the Grouping Order list.
   For example, selecting Analyzer creates a list with top-level nodes that contain the issue category, such as Buffer Overflow, with the issues grouped below by analyzer, such as semantic, or data flow, followed by the issues.
   - Buffer Overflow [0/2]
   --DataFlow [0/1]
   ----Main.cs:234
   ++Semantic [0/1]
6. Repeat Step 5 to select additional grouping types.
7. To change the order of the grouping types:
   a. In the Grouping Order list, select the grouping type that you want to move up or down in the grouping order.
   Right-click the selected grouping type, and then select Move Up or Move Down from the shortcut menu.
About the Source Code Panel

After you open a project in Audit Workbench, the top center panel displays the Project Summary tab. After you select an issue in the issues panel to the left, Audit Workbench adds the source code tab to the top center panel. This source code tab shows the section of code related to the issue selected in the issues panel.

If multiple nodes represent an issue in the Analysis Evidence panel (below the issues panel), the source code tab shows the code associated with the selected node. From the source code tab, you can use the code assist feature to create custom rules and new issues. For information about how to create a new issue from Audit Workbench, see Creating Issues for Undetected Vulnerabilities on page 66.

About Displayed Source Code

After you open an FPR file in Audit Workbench, the source code tab displays source code that is stored locally. If that source code was updated since the last scan, Audit Workbench displays the updated source code, even if the latest scan did not use that updated source code.

However, if that source code is updated after you open the FPR file and Audit Workbench has already started and searched for the source code (even if you close the FPR in Audit Workbench and then re-open it) Audit Workbench does not look for or display the updated source code. It displays the updated source code only after you quit, and then restart Audit Workbench.

About the Project Summary Panel

The Project Summary panel shown in the following screen capture provides detailed information about the scan on the tabs described in the following sections.

To display the Project Summary tab:
1. Open an FPR file.
2. Select Tools → Project Summary.

About the Summary Tab

The Summary tab shows high level information about the project, such as the executable lines of code count.
About the Certification Tab

The **Certification** tab shows the results certification status. Results certification is a check that the analysis has not been altered since it was produced by SCA, PTA, or Runtime Application Protection. Results certification shows specific information about the scanned code, including:

- FPR certification
- Certification details such as the results signature and rules signatures

About the SecurityScope Analysis Tab

The SecurityScope **Analysis** tab shows the following run information:

- The number of issues found by Runtime Application Protection or SecurityScope
- Build ID
- Engine version
- Start and end date and time of the run
- Machine where the run was performed

About the Build Information Tab

The **Build Information** tab shows the following scan information:

- Build details such as the build ID, number of files scanned, and the source last-modified date, which may be different from the date the files were translated
- Executable lines of code scanned
- Total lines of code scanned

This metric includes both executable and non-executable lines of code. This count conforms with non-HP Fortify scan tools.

- List of files scanned with file sizes and timestamps
- Libraries referenced for the scan
- Java classpath used for the translation

About the Analysis Information Tab

The **Analysis Information** tab shows the SCA version, details on computer, user who performed the scan, platform, scan date, machine name, and the elapsed time it took for the code to be scanned in.

The Analysis Information subtabs contain the following information:

- **Rulepacks**—lists the Rulepacks used to scan the source code, including the name, ID, and version number of each Rulepack
- **Properties**—displays the settings of the SCA properties files
- **Command-line Arguments**—displays the command-line options used to analyze the project
- **Warnings**—lists all errors and warnings that occurred during the analysis

To view more information, click the warnings.
About the Functions View

The Functions view in the top right panel shows how and where a function occurs in the source code, whether or not the function was covered by a security rule, and which rule IDs matches the function. The Functions panel can also list the functions that SCA identified as tainted source, and the functions that were not covered by rules in the last scan. For detailed information about the Functions view, see Functions View on page 87.

About the Issue Auditing Panel

The issue auditing panel at the center bottom of the auditing interface provides detailed information about each issue on the tabs described in the following topics.

Note: To show or hide tabs on the Issue Auditing panel, select Options → Show View.

About the Summary Tab

The Summary tab displays information about the selected issue (Table 1) and enables auditors to add comments and custom tag values, and submit the selected issue as a bug.

Table 1: Summary Options

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue</td>
<td>Displays the issue location, including the filename and line number</td>
</tr>
<tr>
<td>Custom Tags area</td>
<td>Displays lists with values that the auditor can add to the issue as attributes For example, valid values for the Analysis tag are as follows:</td>
</tr>
<tr>
<td></td>
<td>• Not an issue</td>
</tr>
<tr>
<td></td>
<td>• Reliability issue</td>
</tr>
<tr>
<td></td>
<td>• Bad practice</td>
</tr>
<tr>
<td></td>
<td>• Suspicious</td>
</tr>
<tr>
<td></td>
<td>• Exploitable</td>
</tr>
<tr>
<td>Suppress Issue</td>
<td>Suppresses the issue</td>
</tr>
<tr>
<td>File Bug</td>
<td>Provides access to a bug-tracking systems such as Bugzilla</td>
</tr>
<tr>
<td>Comments</td>
<td>Appends additional information about the issue to the comment field</td>
</tr>
<tr>
<td>Rule Information</td>
<td>Shows descriptive information such as issue category and kingdom</td>
</tr>
<tr>
<td>More Information</td>
<td>Displays the Details tab</td>
</tr>
<tr>
<td>Recommendations</td>
<td>Displays the Recommendations tab</td>
</tr>
</tbody>
</table>
### About the SecurityScope Details Tab

The Details tab provides a detailed description of the selected issue and offers guidelines for addressing it. Each description includes some or all of the sections described in Table 2.

**Table 2: Details Tab**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract/Custom Abstract</td>
<td>Summary description of an issue, including any custom abstracts defined for your organization</td>
</tr>
</tbody>
</table>
| Explanation/Custom Explanation | Description of the conditions in which this type of issue occurs  
Includes a discussion of the vulnerability, the constructs typically associated with it, how it can be exploited, and the potential ramifications of an attack.  
Also provides custom explanations defined for your organization. |
| Instance ID               | Provides a unique identifier for the issue                                                                                                                    |
| Rule ID                   | Identifies the primary rule that uncovered the issue                                                                                                          |
| Priority Metadata Values  | Includes IMPACT and LIKELIHOOD                                                                                                                             |
| Legacy Priority Metadata Values | Includes SEVERITY and CONFIDENCE                                                                           |

The SecurityScope Details tab shows the following information about runtime issues found by PTA or Runtime Application Protection:

**Arguments**—shows the Index, Value, and Return Value

**Request**—the HTTP Request information including the Method URL, content-length, accept-encoding, referer, connection, accept-language, host, accept-charset, user-agent, content-type, cookie, accept, keep-alive

**Stack Trace**—shows the order of methods called during execution and line number information. Blue code links are clickable and only display for code scanned by SCA

### About the Recommendations Tab

The Recommendations tab contains suggestions and examples of how to secure the vulnerability or remedy the bad practice.

**Table 3: Recommendations Tab**

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recommendations/Custom Recommendations</td>
<td>Provides recommendations for this type of issue, including examples, as well as custom recommendations defined by your organization</td>
</tr>
<tr>
<td>Tips/Custom Tips</td>
<td>Provides tips for this type of issue, including any custom tips defined by your organization</td>
</tr>
<tr>
<td>References/Custom References</td>
<td>Provides reference information, including any custom reference defined by your organization</td>
</tr>
</tbody>
</table>
About the History Tab
Shows a complete list of auditing actions, including details such as the time and date, and the name of the user who modified the issue.

About the Diagram Tab
The Diagram tab illustrates the node execution order, call depth, and expression type of the issue selected in the issues panel.

Example of a Blind SQL Injection Issue Data Flow Diagram
![Diagram of Blind SQL Injection Issue](image)

The Diagram tab displays information relevant to the rule type. Execution order is represented along the vertical axis.

For dataflow issues, the trace starts at the top with the first function to call the taint source, then traces the calls to the source (blue node), and ends the trace at the sink (red node). In the diagram, the source (src) and sink nodes are also labeled. A red “X” on a vertical axis indicates that the function called finished executing.

The horizontal axis shows the call depth. A line shows the direction that control is passed. If control passes with tainted data traveling through a variable the line is red, and when it is without tainted data, the line is black.

The icons used for the expression type of each node in the diagram are the same icons as those used in the Analysis Evidence panel. To view the icons and the descriptions, see About the Analysis Evidence Panel on page 42.

About the Filters Tab
The Filters tab displays all the filters that are in the selected filter set.

Table 4: Filters Tab Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filters</td>
<td>Displays a list of the visibility and folder filters configured in the selected filter set.</td>
</tr>
<tr>
<td></td>
<td>• Visibility filters show or hide issues</td>
</tr>
<tr>
<td></td>
<td>• Folder filters sort the issues into the folders in the analysis results panel</td>
</tr>
<tr>
<td></td>
<td>Right-click a filter to show issues that match the filter or to enable, disable, copy, or delete it.</td>
</tr>
</tbody>
</table>
**Table 4: Filters Tab Options**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **If** | Displays the filters conditions  
The first list displays issue attributes, the second list specifies how to match the attribute, and third shows the value the filter matches.  
**Note:** This option is visible when you create a new filter or edit an existing filter. In this case, a dialog box displays the **If** section. |
| **Then** | Indicates the filter type, where *hide* is a visibility filter and *folder* is a folder filter  
**Note:** This option is visible when you create a new filter or edit an existing filter. In this case, a dialog box displays the **Then** section. |
About the Analysis Evidence Panel

When you select an issue in the issues panel, the **Analysis Evidence** panel displays the evidence that the analyzer used to identify the issue. Evidence is presented in the order it was discovered. For dataflow issues, this evidence represents the path that the tainted data follows from the source function to the sink function.

For example, if you select an issue related to a potentially tainted data flow, the **Analysis Evidence** panel shows the direction in which the data flow is moving in this section of the source code.

Table 5 lists the icons used in the **Analysis Evidence** panel to show how data flow moves in the source code.

**Table 5: Analysis Evidence Icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Icon 1" /></td>
<td>Data are assigned to a field or variable</td>
</tr>
<tr>
<td><img src="image2.png" alt="Icon 2" /></td>
<td>Information is read from a source external to the code (html form, url, and so on)</td>
</tr>
<tr>
<td><img src="image3.png" alt="Icon 3" /></td>
<td>Data are assigned to a globally scoped field or variable</td>
</tr>
<tr>
<td><img src="image4.png" alt="Icon 4" /></td>
<td>Comparison is made</td>
</tr>
<tr>
<td><img src="image5.png" alt="Icon 5" /></td>
<td>Function call receives tainted data</td>
</tr>
<tr>
<td><img src="image6.png" alt="Icon 6" /></td>
<td>Function call returns tainted data</td>
</tr>
<tr>
<td><img src="image7.png" alt="Icon 7" /></td>
<td>Passthrough, tainted data passes from one parameter to another in a function call</td>
</tr>
<tr>
<td><img src="image8.png" alt="Icon 8" /></td>
<td>An alias is created for a memory location</td>
</tr>
<tr>
<td><img src="image9.png" alt="Icon 9" /></td>
<td>Data are read from a variable</td>
</tr>
<tr>
<td><img src="image10.png" alt="Icon 10" /></td>
<td>Data are read from a global variable</td>
</tr>
<tr>
<td><img src="image11.png" alt="Icon 11" /></td>
<td>Tainted data is returned from a function</td>
</tr>
</tbody>
</table>
Table 5: Analysis Evidence Icons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;</td>
<td>A pointer is created</td>
</tr>
<tr>
<td>*</td>
<td>A pointer is dereferenced</td>
</tr>
<tr>
<td>...x</td>
<td>Scope of a variable ends</td>
</tr>
<tr>
<td>¬</td>
<td>Execution jumps</td>
</tr>
<tr>
<td></td>
<td>Branch in the code’s execution</td>
</tr>
<tr>
<td>!x</td>
<td>A branch is not taken in the code’s execution</td>
</tr>
<tr>
<td></td>
<td>Generic</td>
</tr>
<tr>
<td><img src="icon.png" alt="icon" /></td>
<td>A runtime source, sink, or validation step</td>
</tr>
</tbody>
</table>

The **Analysis Evidence** panel can contain inductions, which provide supporting evidence for their parent nodes. Inductions consist of a text node, displayed in italics as a child of the trace node, and an induction trace, which is displayed as a child of the text node. A box surrounds the induction trace. The italics and the box distinguish the induction from a standard subtrace.
Customizing the Auditing Interface

You can modify your interface preferences to specify how issues are listed in the analysis results panel.

To customize the issues list:

1. Select **Options → Options**.
2. In the left panel of the Options dialog box, select **Interface Preferences**.
3. Select **HP Fortify Software → Options**.
   
   The Options dialog box opens.
4. In the left panel, select **Interface Preferences**.
   
   The following table describes the interface preferences listed in the right panel.

<table>
<thead>
<tr>
<th>Preference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show Suppressed Issues</td>
<td>Shows all suppressed issues (disabled by default)</td>
</tr>
<tr>
<td>Show Removed Issues</td>
<td>Shows all issues that were uncovered in the previous analysis, but are no longer evident in the new SCA analysis results.</td>
</tr>
<tr>
<td></td>
<td>When multiple scans are run on a project over time, vulnerabilities are often remediated or become obsolete. Static Code Analysis marks these vulnerabilities as Removed Issues.</td>
</tr>
<tr>
<td>Show Hidden Issues</td>
<td>Shows all hidden items</td>
</tr>
<tr>
<td>Collapse Issues</td>
<td>Shows similar issues based on certain attributes under a shared parent node in the Issues tree</td>
</tr>
<tr>
<td>Use Short File Names</td>
<td>References the issues in the Issues view by file name only, instead of by relative path (enabled by default)</td>
</tr>
<tr>
<td>Show Category of Issue</td>
<td>Shows the category to which each issue belongs after the file name and line number</td>
</tr>
<tr>
<td>Show Only My Issues</td>
<td>Displays the name text on the folder tabs</td>
</tr>
<tr>
<td>Show Abstract in Issue Summary</td>
<td>Shows the abstract information on the right of the Summary tab</td>
</tr>
<tr>
<td>Show Comments in Issue Summary</td>
<td>Shows comments in the center of the Summary tab</td>
</tr>
<tr>
<td>Show ‘All’ Folder in Issue Summary Graph</td>
<td>Shows another bar in the chart on the Project Summary tab</td>
</tr>
<tr>
<td>Right justify ‘All’ folder</td>
<td>Displays the All folder with its contents listed on the right</td>
</tr>
<tr>
<td>Display name in folder tabs</td>
<td>Displays the name text in the folder tabs</td>
</tr>
<tr>
<td>Include Comments in History view</td>
<td>Shows the history items for comments on the History tab</td>
</tr>
<tr>
<td>Attribute Used for Quick Audit Action</td>
<td>From this list, you can select a custom tag for quick audit actions. This enables you to assign custom tag values to issues. For information about quick audits, see Performing Quick Audits on page 64.</td>
</tr>
</tbody>
</table>

5. To specify your interface preferences, select or clear the preference check boxes.

   **Note:** To restore the default settings at any time, click **Reset Interface**.

6. To save your preferences, click **OK**.
About Searching Issues

After scan results are uploaded to Audit Workbench, you can use the search box at the bottom of the issues panel to find specific issues and to limit the issues displayed in a folder. After you type a search term, the label next to the folder name changes to indicate the number of issues that match the search as a subset of the total.

You can wrap search terms with delimiters to indicate the type of comparison to be performed. Table 6 shows the syntax to use in the search string field.

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>contains</td>
<td>Searches for a term without any qualifying delimiters</td>
</tr>
<tr>
<td>equals</td>
<td>Searches for an exact match if the term is wrapped in quotation marks (&quot;&quot;&quot;)</td>
</tr>
<tr>
<td>regex</td>
<td>Searches for values that match a Java-style regular expression delimited by a forward slash (/)</td>
</tr>
<tr>
<td>number range</td>
<td>Uses standard mathematical syntax, such as &quot; (&quot;and&quot;) &quot; for exclusive range, and &quot; (&quot;and&quot;) [&quot; for inclusive range, where (2,4] represents the range of numbers greater than two, and less than or equal to four</td>
</tr>
<tr>
<td>not equals</td>
<td>Excludes issues specified by the string by preceding the string with an exclamation character (!) For example, file:!Main.java returns all issues that are not in the Main.java file.</td>
</tr>
</tbody>
</table>

Search terms can be further qualified with modifiers. For more information, see About Search Modifiers on page 46. The basic syntax for using a modifier is modifier:<search_term>.

A search string can contain multiple modifiers and search terms. If you specify more than one modifier, the search returns only issues that match all the modified search terms. For example, file:ApplicationContext.java category:SQL Injection returns only SQL injection issues found in ApplicationContext.java.

If you use the same modifier more than once in a search string, then the search terms qualified by those modifiers are treated as an OR comparison. So, for example, file:ApplicationContext.java category:SQL Injection category:Cross-Site Scripting returns SQL injection issues and cross-site scripting issues found in ApplicationContext.java.

For complex searches, you can also insert the AND or the OR keyword between your search queries. Note that AND and OR operations have the same priority in searches.
About Search Modifiers

You can use a search modifier to specify which issue attribute the search term should apply to. To use a modifier that contains a space in the name, such as the name of the custom tag, you must delimit the modifier with brackets. For example, to search for issues that are new, type \[issue \text{ age}\]:new.

A search that is not qualified by a modifier matches the search string on the following attributes: kingdom, primary rule id, analyzer, filename, severity, class name, function name, instance id, package, confidence, type, subtype, taint flags, category, sink, and source.

- To apply the search to all modifiers, enter a string, such as \textit{control flow}. This searches all of the modifiers and returns any results that contain the string “control flow.”
- To apply the search to a specific modifier, type the modifier name and the string as follows: analyzer:control flow. This returns all results with the analyzer “control flow.”

Table 7 describes the search modifiers.

\begin{table}[h]
\centering
\begin{tabular}{|l|p{15cm}|}
\hline
\textbf{Modifier} & \textbf{Description} \\
\hline
\[issue \text{ age}\] & Searches for the issue age, which is either \texttt{removed}, \texttt{existing}, or \texttt{new} \\
\hline
<custom_tagname> & Searches the specified custom tag. Note that tag names that contain spaces must be delimited by square brackets. Example: [my tag]:value \\
\hline
analysis & Searches for issues that have the specified audit analysis value (such as "exploitable," "not an issue," and so on) \\
\hline
analyzer & Searches the issues for the specified analyzer \\
\hline
audience & Searches for issues by intended audience. Valid values are “targeted,” “medium,” and “broad” \\
\hline
audited & Searches the issues to find \texttt{true} if Primary Custom Tag is set and \texttt{false} if Primary Custom Tag is not set \\
\hline
category (cat) & Searches for the given category or category substring \\
\hline
comments (comment, com) & Searches the comments submitted on the issue \\
\hline
commentuser & Searches for issues with comments from a specified user \\
\hline
confidence (con) & Searches for issues that have the specified confidence value. Fortify Source Code Analyzer calculates the confidence value based on the number of assumptions made in code analysis. The more assumptions made, the lower the confidence value. \\
\hline
dynamic & Searches for issues that have the specified dynamic hot spot ranking value \\
\hline
file & Searches for issues where the primary location or sink node function call occurs in the specified file. \\
\hline
\end{tabular}
\end{table}
Chapter 3: Audit Workbench Projects

Searches for issues that have a priority level that matches the specified priority determined by the HP Fortify analyzers. Valid values are critical, high, medium, and low, based on the expected impact and likelihood of exploitation.

The impact value indicates the potential damage that might result if an issue is successfully exploited. The likelihood value is a combination of confidence, accuracy of the rule, and probability that the issue can be exploited.

Audit Workbench groups issues into folders based on the four priority values (critical, high, medium, and low) by default.

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[fortify priority order]</td>
<td>Searches for issues that have a priority level that matches the specified priority determined by the HP Fortify analyzers. Valid values are critical, high, medium, and low, based on the expected impact and likelihood of exploitation. The impact value indicates the potential damage that might result if an issue is successfully exploited. The likelihood value is a combination of confidence, accuracy of the rule, and probability that the issue can be exploited. Audit Workbench groups issues into folders based on the four priority values (critical, high, medium, and low) by default.</td>
</tr>
<tr>
<td>historyuser</td>
<td>Searches for issues that have audit data modified by the specified user.</td>
</tr>
<tr>
<td>kingdom</td>
<td>Searches for all issues in the specified kingdom.</td>
</tr>
<tr>
<td>maxconf</td>
<td>Searches for all issues that have a confidence value up to and including the number specified as the search term.</td>
</tr>
<tr>
<td>&lt;metagroup_name&gt;</td>
<td>Searches the specified metagroup. Metagroups include [owasp top ten 2010], [sans top 25 2010], and [pci 2.1], and others. Square braces delimit field names that include spaces.</td>
</tr>
<tr>
<td>minconf</td>
<td>Searches for all issues that have a confidence greater than or equal to the specified value.</td>
</tr>
<tr>
<td>package</td>
<td>Searches for issues where the primary location occurs in the specified package or namespace. (For data flow issues, the primary location is the sink function.)</td>
</tr>
<tr>
<td>[primary context]</td>
<td>Searches for issues where the primary location or sink node function call occurs in the specified code context. Also see sink, [source context].</td>
</tr>
<tr>
<td>primaryrule (rule)</td>
<td>Searches for all issues related to the specified sink rule.</td>
</tr>
<tr>
<td>ruleid</td>
<td>Searches for all issues reported by the specified rule IDs used to generate the issue source, sink and all passthroughs.</td>
</tr>
<tr>
<td>sink</td>
<td>Searches for issues that have the specified sink function name. Also see [primary context].</td>
</tr>
<tr>
<td>source</td>
<td>Searches for data flow issues that have the specified source function name. Also see [source context].</td>
</tr>
<tr>
<td>[source context]</td>
<td>Searches for data flow issues that have the source function call contained in the specified code context. Also see source, [primary context].</td>
</tr>
<tr>
<td>sourcefile</td>
<td>Searches for data flow issues with the source function call that the specified file contains. Also see: file.</td>
</tr>
<tr>
<td>status</td>
<td>Searches issues that have the status reviewed, not reviewed, or under review</td>
</tr>
<tr>
<td>suppressed</td>
<td>Searches for suppressed issues.</td>
</tr>
<tr>
<td>taint</td>
<td>Searches for issues that have the specified taint flag.</td>
</tr>
<tr>
<td>trace</td>
<td>Searches for issues that have the specified string in the data flow trace.</td>
</tr>
</tbody>
</table>

Table 7: Search Modifiers
Table 7: Search Modifiers

<table>
<thead>
<tr>
<th>Modifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tracenode</td>
<td>Enables you to search on the nodes within an issue’s analysis trace. Each tracenode search value is a concatenation of the tracenode’s file path, line number, and additional information.</td>
</tr>
<tr>
<td>&lt;no attribute&gt;</td>
<td>Searches for issues that have any of the most common attributes that match the specified string</td>
</tr>
</tbody>
</table>

Search Query Examples

Consider the following examples:

- To search for all privacy violations in file names that contain jsp with getSSN() as a source, type the following:
  
  category:"privacy violation" source:getssn file:jsp

- To search for all file names that contain com/fortify/awb, type the following:
  
  file:"com/fortify/awb"

- To search for all paths that contain traces with mydbcode.sqlcleanse as part of the name, type the following:
  
  trace:mydbcode.sqlcleanse

- To search for all paths that contain traces with cleanse as part of the name, type the following:
  
  trace:cleanse

- To search for all issues that contain cleanse as part of any modifier, type the following:
  
  cleanse

- To search for all suppressed vulnerabilities with asdf in the comments, type the following:
  
  suppressed:true comments:asdf

- To search for all categories except for SQL Injection, type the following:
  
  category:!:SQL Injection
Performing Simple Searches

To use the search box to perform a simple search, do one of the following:

- Type a search string in the box and press ENTER.

Alternatively,

- To select a search term you used previously, click the arrow in the search box, and then select a search term from the list.

To get assistance in composing the comparison for your search string, do the following:

1. Click your cursor in the search box, and then press CTRL + SPACE.

2. From the displayed list, double-click an issue attribute to begin your search string.

3. To get assistance specifying the comparison, with your cursor placed after the modifier in the search box, press CTRL + SPACE.

4. From the displayed list, double-click the comparison to add to your search string.

5. Finish typing the search term.

The issues panel lists all of the issues that match your search string.

Audit Workbench saves all of the search terms you enter for the current session. To select a search term you used previously, click the arrow in the search box, and then select a search term. (After you quit Audit Workbench, the saved search terms are discarded.)

Creating complex search strings can involve several steps. If you enter an invalid search string, the magnifying glass icon in the text field changes to a warning icon to notify you of the error. Click the warning sign to view information about the search term error.

The advanced search feature makes it easier to build complex search strings. For a description of this feature and instructions on how to use it, see Performing Advanced Searches.
Performing Advanced Searches

You can use the advanced search feature to build complex search strings.

To use the advanced search feature:

1. In the search box, type a search term.
2. To the right of the search box, click Advanced.

Audit Workbench parses the search term, and uses it to populate the Advanced Search dialog box. The box on the left displays the modifier, the middle box displays the comparison and type, and the box on the right displays the search term.

Audit Workbench adds a new AND query row to the dialog box.

3. To change the AND operator to an OR operator, double-click AND / OR.
4. Select the modifier, the comparison and type, and the search term from the lists.
   The list for the search term includes the known values in the current scan for the specified attribute. However, you can type any value into this field. To specify an unqualified search term, select Any Attribute from the bottom of the modifier list.

5. To add an AND query row, in the top right corner of the dialog box, click AND add. To add an OR query row, in the top right corner of the dialog box, click OR add.
6. Add as many query rows as you need for the search.
7. To delete a row, to the right of the row, click Delete. To remove all rows, click Clear.
8. Click Find.

Note: As you build your search string, the Advanced Search dialog box displays any errors in the status below the search string builder. The Find button is not enabled unless all errors are resolved.

Consider the following examples:

- To search for all privacy violations in file names that contain jsp with getSSN() as a source, type the following:
  category: "privacy violation" source:getssn file:jsp

- To search for all file names that contain com/fortify/awb, type the following:
  file: "com/fortify/awb"

- To search for all paths that contain traces with mydbcode.sqlcleanse as part of the name, type the following:
  trace: mydbcode.sqlcleanse

- To search for all paths that contain traces with cleanse as part of the name, type the following:
  trace: cleanse

- To search for all issues that contain cleanse as part of any modifier, type the following:
  cleanse

- To search for all suppressed vulnerabilities with asdf in the comments, type the following:
  suppressed: true comments: asdf
• To search for all categories except for SQL Injection, type the following:

\texttt{category:!SQL~Injection}

\textbf{About Custom Tags}

In Software Security Center, code auditing involves the security team's examining HP Fortify project scan results (FPR) and assigning values to custom tags associated with project issues. The development team can then use these tag values to determine which issues to address and in what order.

To enable project auditing out of the box, Software Security Center provides a single default tag named “Analysis.” Valid values for the Analysis tag are Exploitable, Not an Issue, Suspicious, Reliability Issue, and Bad Practice. You can modify the Analysis tag attributes, revise the tag values, or add new values based on your auditing needs.

To refine your auditing process, you can define your own custom tags. For example, you could create a custom tag that can be used to track the sign-off process for an issue. After a developer audits his own issues, a security expert can review those same issues and mark each as “approved” or “not approved.”

You can also define custom tags from Software Security Center, either directly with project template uploads through Software Security Center, or from Audit Workbench through project templates in FPR files.

\textbf{Note:} Although you can add new custom tags from Audit Workbench (AWB) as you audit a project, if these custom tags are not defined in Software Security Center for the project template associated with the project version, then the new tags are lost if you upload the FPR file to Software Security Center.

For more information about custom tags, see the \textit{HP Fortify Software Security Center User Guide}.

You can add the following attributes to your custom tags:

• Extensible—This enables users to create a new value while auditing, even without the permission to manage custom tags.

• Restriction—This restricts who can set the tag value on an issue. Administrators, security leads, and managers have permission to audit restricted tags.

• Hidden (Software Security Center only)—This setting is used to hide a tag from a project version or project template.
Adding Custom Tags: For FPRs Not Uploaded to Software Security Center

If you open analysis results (either by opening an FPR file, or scanning a project) that have not been uploaded to Software Security Center, you can create a custom tag that can be used to audit the results, and specify its values and attributes.

To create a custom tag for results that have not been uploaded to Software Security Center:

1. Select **Tools → Project Configuration**. The Project Configuration dialog box opens.
2. Click the **Custom Tags** tab.
3. At the top of the **Tags** panel, click the plus icon ( ).
4. In the Enter Value dialog box, type a label for the new tag, and then click **OK**.
5. To enable users to add new values for the tag during an audit, leave the **Extensible** check box on the **Custom Tags** tab selected. To enable only managers, security leads, and administrators to add new values for the tag during an audit, clear the check box.
6. To specify a value for the custom tag:
   a. At the top of the **Values** column, click the plus icon ( ).
   b. In the Enter Value dialog box, type a value for the new tag, and then click **OK**.
   c. Repeat **Step a** and **Step b** for as many values as you need for the tag.

The value that you give your custom tag can be a discreet attribute for the particular issue this custom tag addresses. For example, you may want to specify that this custom tag addresses a due date or server quality issue.

7. (Optional) Enter descriptions of the custom tag and its values in the corresponding **Description** boxes.
8. (Optional) From the **Default Value** list, select the default value for the tag.

   If you set a default value for the tag, then issues that do not have a value set for that tag have the default value. If no default value is specified, then the tag value is set to **Not Set**.
9. Click **OK**.

The **Summary** tab now displays the new tag and its default value (if you assigned one).

**Committing Custom Tags: For FPRs Not on Software Security Center**

To commit a custom tag to Software Security Center for an FPR that has not been on Software Security Center:

1. Highlight the custom tag.
2. Click **Commit**.

3. Provide your Software Security Center username and password, and then click **OK**.

   If the project template is not on Software Security Center, the Custom Tag Upload dialog box opens.

4. Do one of the following:
   
   - To have Audit Workbench upload custom tags to the global pool on Software Security Center, click **Yes**.
   - To prevent Audit Workbench from uploading custom tags to the global pool on Software Security Center, click **No**.

   The modified custom tags are also updated in the global pool.

Custom tags are not removed from the global pool. In Audit Workbench you are always looking at the custom tags for a project. These are a subset of tags in the global pool. This ensures that custom tags not visible in Audit Workbench can still be used for other projects.
Synchronizing Custom Tags—For an FPR Not on Software Security Center

To synchronize a custom tag for an FPR that has not been on Software Security Center (and where the project template has), use the following procedure. In this instance synchronizing custom tags retrieves the custom tags from Software Security Center.

1. Highlight the custom tag.
2. Click **Synchronize**.
   
   The SSC Login dialog box opens.
3. Provide your Software Security Center credentials, and then click **OK**.
   
   The Custom Tag Download dialog box opens.
4. Do one of the following:
   
   - To download the custom tags from the project template (a set of custom tags retrieved from the global pool on Software Security Center), click **Yes**.
   - To cancel the operation, click **No**.

Auditing Custom Tags: For FPRs on Software Security Center

To audit a custom tag for an FPR that has been on Software Security Center:

1. Select **Tools** → **Project Configuration**.
   
   The Verify Collaborative Project Permissions dialog box opens.
2. Click **Verify**.
   
   Permissions on Software Security Center are selected. If you click **Cancel**, the Filter Sets, Folders, and Custom Tags tabs in the Project Configuration dialog box are disabled.
3. Enter your user name and password, and then click **OK**.
   
   - If you do not have permissions on Software Security Center, all restricted tag combination boxes are permanently disabled for the remainder of the Audit Workbench session.
   - If you have permissions on Software Security Center, all restricted custom tag combination boxes are permanently enabled for the remainder of the Audit Workbench session and you may set restricted tag values similar to the method you use to set unrestricted tag values.

   **Note:** The **Import** button on the Filter Sets tab is disabled to prevent you from importing a new local project template for an FPR that has been on Software Security Center.

Committing Custom Tags—For an FPR on Software Security Center

To commit custom tags for an FPR that has been uploaded to Software Security Center, do the following:

1. Select **Tools** → **Project Configuration**.
   
   The Project Configuration dialog box opens.
2. Click the Custom Tags tab.
3. Highlight the custom tag.
4. Click **Commit**.
   
   The Custom Tag Upload dialog box opens.
5. Do one of the following:
   
   - To upload the custom tags to the global pool without assigning them to the project version, click **No**.
   - To upload the custom tags to the global pool and assign them to the project version (but not the project template) click **Yes**.
Synchronizing Custom Tags—For an FPR on Software Security Center

To synchronize custom tags for an FPR that has been uploaded to Software Security Center.

1. Select **Tools → Project Configuration**, and then click the **Custom Tags** tab.
2. Highlight the custom tag.
3. Click **Synchronize**.
   The Custom Tag Download dialog box opens.
4. To download custom tags from the project template, click **Yes**.

Committing Filter Sets and Folders

If the FPR has been uploaded to Software Security Center, and you want to commit filter sets or folders, do the following.

1. Select **Tools → Project Configuration**.
   The Project Configuration dialog box opens.
2. Click the **Filter Sets** (or **Folders**) tab.
3. Select the filter set (or folder) from the list.
4. Click **Commit**.
   The SSC Login dialog box opens.
5. Log on to your Software Security Center instance.
   The Update Existing Project Template dialog box opens.
6. To upload filter sets and folders to the project template, click **Yes**.
   If the FPR has not been uploaded to Software Security Center, the Add Project Template dialog box opens.
7. To add the project template file that contains the current set of custom tags to Software Security Center, click **Yes**.
Synchronizing Filter Sets and Folders

If the FPR has been on Software Security Center, use the following procedure to download filter sets and folders configured for the project on Software Security Center.

1. Select **Tools → Project Configuration**.
2. Click the **Filter Sets** tab.
3. Click **Synchronize**.

   Audit Workbench displays a message to advise you that downloading filter sets and folders from Software Security Center overwrites your local filter sets and folders.

4. To proceed with the synchronization, click **Yes**.

   The SSC Login dialog box opens.

5. Provide your Software Security Center credentials, and then click **OK**.

   The Project Templates dialog box opens.

6. In the **Project Template** column, select a project template name.
7. Click **OK**.

Audit Workbench downloads the filter sets and folders from the selected project template on Software Security Center.

**About Project Template Selection**

Audit Workbench applies project template precedence to projects in the following order:

- Template in the FPR
- Template in `<install root>/Core/config/filters/defaulttemplate.xml`
- Template in `<install root>/Core/config/rules/defaulttemplate.xml` or `projecttemplate`
- Embedded HP Fortify default template

After you associate a project template with a project, Audit Workbench applies all changes that you make to the project (such as the addition of folders, custom tags, filter sets, or filters) to the associated project template. For information about how to change the project template associated with a project, see *Importing Project Templates* on page 57.
About Sharing Project Templates with Software Security Center

This section covers the following topics:

- Exporting Project Templates
- Importing Project Templates
- Downloading Project Templates from Software Security Center
- Uploading Project Templates to Software Security Center

Exporting Project Templates

Exporting a project template creates a file that contains the current project’s filter sets and custom tags. After you export a project template, you can import it into another project file.

To export a project template:

1. Select **Tools → Project Configuration**.
   
   The Project Configuration dialog box opens.

2. Click the **Filter Sets** tab.

3. Click **Export**.
   
   The Select Template File Location dialog box opens.

4. Browse to the location where you want to save the file.

5. Enter a file name without an extension.

6. Click **Save**.

Audit Workbench saves the current settings of the template to the new XML file.

Importing Project Templates

Importing a project template overwrites the project configuration settings. The local filter sets and custom tags are replaced with the filter sets and custom tags in the project template.

To import a project template:

1. Select **Tools → Project Configuration**.
   
   The Project Configuration dialog box opens.

2. Click the **Filter Sets** tab.

3. Click **Import**.
   
   The Locate Template File dialog box opens.

4. Select the project template file to import.

5. Click **Import**.

Audit Workbench updates the filter sets, folders, and custom tags.

To return the settings to the default project template settings, click **Reset to Default**.

Downloading Project Templates from Software Security Center

You can use Audit Workbench to download project templates from Software Security Center. After you download a template, the imported project template is applied to the open project.

To download a project template:

1. Select **Tools → Project Configuration**.
The Project Configuration dialog box opens to the Filter Sets tab.

2. Click Synchronize.

If you have not logged on to Software Security Center from Audit Workbench, the Software Security Center login dialog box opens.

3. If necessary, provide your Software Security Center logon credentials, and then click OK.

The Project Templates dialog box opens lists the project templates available on Software Security Center.

4. From the Project Template column, select the template to download.

5. Click OK.

Audit Workbench imports the template and associates it with the open project.

The Project Template Name box on the Filter Sets tab displays the template name.

**Uploading Project Templates to Software Security Center**

You can use Audit Workbench to upload projects to Software Security Center.

To upload a project template:

1. Select Tools → Project Configuration.

The Project Configuration dialog box opens.

2. Click Commit.

3. When prompted enter the user name and password for the Software Security Center from which you want to download a project template.

The Add Project Template dialog box opens. Audit Workbench prompts you to either add the project template or to overwrite an existing project template that has the same ID.
About Working with Audit Data

This section contains the following topics:

- Opening Projects
- About Merging Audit Data
- About the Event Bridge Utility
- About Additional Metadata
- Uploading Results to Software Security Center

Opening Projects

To open an FPR file:

1. Open Audit Workbench.
2. Select File → Open Project.
   
   The Choose Project dialog box opens.
3. Browse to and select the FPR file, and then click Open.

If the FPR format is HP Fortify Static Code Analyzer version 4.5.1 or earlier, the Migration Wizard automatically starts. Otherwise, the FPR displays in the Audit perspective. Migrate the issue IDs before you migrate the audit data, as described in the HP Fortify Static Code Analyzer Migration Guide.

About Merging Audit Data

You can merge audit data into your project from another file. Audit data consist of the custom tag values and comments added to issues from Audit Workbench. Audit Workbench merges comments in a chronological list as it updates custom tag values. If the custom tag values conflict (the same tag is set to different values) Audit Workbench prompts you to resolve the conflict.

Note: Keep in mind that Audit Workbench:

- Does not merge issues
- Shows only the most recent scan
- Shows only the issues from the most recent scan

Ensure that the projects you merge contain the same analysis information, that is the scan was on the same source code project (no missing libraries or files), the SCA settings were the same, and the scan was performed using the same security content.

Merging Using

To merge projects:

1. Open a project in Audit Workbench.
2. Click Tools → Merge Audit Projects.
   
   The Choose an SCA Analysis File dialog box opens.
3. Select an FPR file, and then click Open.
   
   The Merge dialog box opens.

   Note: After you select an FPR, Audit Workbench might prompt you to choose between the project template in the current FPR and the project template in the FPR you are merging in.

4. To confirm the number of issues added or removed from the file, click OK.

   Note: If the scan is identical, Audit Workbench does not add or remove issues.
About Merging Audit Data Using the HP Fortify Audit Project Command-line Utility

You can also use the HP Fortify Audit Project Command-line Utility (bin\FPRutility.bat for Windows) to merge audit data. This utility enables you to merge an audited project, verify the signature of the FPR, or migrate 4.x HP Fortify Audit Projects to the current format.

Migration for 5.x projects occurs automatically with the merge action. Merging combines all analysis information resolving conflicts using the values set in the primary project. The merge produces an output project file that contains the analysis information from the primary project.

The signature action prints the full signature information. Exit codes are used to relay the validity of the signature.

About the Event Bridge Utility

The Event Bridge converts an event log into an FPR. Events logs contain runtime events generated by HP Fortify’s dynamic analysis products.

Whenever you open (File → Open Project) or merge (Tools → Merge Audit Projects) an event log, the Event Bridge utility is executed in the background. The result is an FPR that contains a runtime FVDL. This runtime FVDL is populated with runtime issues, each of which contains information for one or more events from the event log. Audit Workbench then opens the converted runtime FPR, making the entire process transparent. Once the converted FPR is opened or merged, you can save it to a location of your choice. Reopening this saved FPR opens the FPR directly; the Event Bridge is no longer executed because the event log has already been converted.

As an alternative, you can use the Event Bridge directly and then open the converted FPR result in Audit Workbench. This is the same process as opening up the event log in Audit Workbench, except that the Event Bridge is executed manually instead of automatically. For information about how to run the Event Bridge using the event2fpr utility, see the HP Fortify SecurityScope Installation Guide.

About Additional Metadata

Each issue in Audit Workbench contains additional metadata that is not produced by HP Fortify’s internal analyzers. Examples include alternative categories (for example, OWASP, CWE, WASC), and prioritization values that are used in the default filters (for example, impact, accuracy, probability). These metadata attributes are viewable through the standard Group By and Search mechanisms.

If you open an older FPR that does not contain metadata values, the metadata values for the issues are retrieved from legacy mapping files. These legacy mapping files exist in the Core\Config\LegacyMappings directory, and are indexed by either issue category, or issue category and analyzer. The legacy mapping files are accessed as needed, so each issue in your project must always have metadata values, whether those values come from the FPR, the legacy mapping files, or a combination of the two.

Uploading Results to Software Security Center

You can upload results to Software Security Center.

To upload results to Software Security Center:

1. Select Options → Options.
   - The Options dialog box opens.
2. Click Server Configuration.
3. In the Software Security Center Configuration section, specify Server URL for Software Security Center configuration, such as http://111.0.0.1:8181.
4. If necessary, specify a proxy server and port.
5. Click OK.

7. Select **Tools → Upload Audit Project**.
   A dialog box lists the current projects.

8. Select a project, and then click **OK**.

Audit Workbench displays a message after the upload is completed.
About Advanced Configuration

This section contains the following topics:

- Using the Audit Guide in Advanced Mode
- About Integrating with Bug Tracking Systems
- About Third-Party Results
- About Public APIs
- About Penetration Test Schema

Using the Audit Guide in Advanced Mode

You can use the Audit Guide during your issues audit. Its advanced features enable you to determine how issues are filtered in the results.

To use the Audit Guide:

1. Select **Tools → Audit Guide**.
   
   The Audit Guide wizard starts.
2. Click **Advanced Mode**.
3. In the **Audit Guide Questions** list, select the types of issues to filter out. (Click an issue type to see a description of the issue type in Audit Workbench.)
   
   When you select items in the Audit Guide Questions list, Audit Workbench displays the filter details for this issue type at the bottom of the dialog box and shows the number of issues found by each filter.
4. Click **OK**.

Audit Workbench applies the filters to the issues.

About Integrating with Bug Tracking Systems

Audit Workbench provides a plug-in interface for integrating with defect tracking systems so that you can file bugs directly from Audit Workbench. To file a bug, right-click an issue and select **File Bug** to choose from a list of previously configured bug tracking plug-ins to use to file the bug.

An example plug-in is provided for Bugzilla (www.bugzilla.org).

Source code for the Bugzilla plug-in is available in the following directory:

<fortify_install>/Samples/advanced/Bugzilla

As delivered, Audit Workbench supports integration with Bugzilla 3.4 and ALM 11/ALI 2.0 bug tracking systems. If your company uses a different bug tracking system, then you can author a new plug-in for it. For instructions, see the README.txt file, which is located in the <fortify_install>/Samples/advanced/Bugzilla directory.

**Important:** If your custom bug tracker accesses supporting jar files, you must add them to the **Class-Path** attribute in your bug tracker's MANIFEST.MF file.
About Third-Party Results

HP Fortify ships built-in parsers for the following penetration test (pentest) tools and services:

- HP WebInspect
- IBM AppScan
- Application Security Inc. AppDetective
- White Hat

Each of the above tools and services produces results in an XML file format. The built-in HP Fortify parsers take in a pentest results file as input and dynamically create HP Fortify issues.

The parsers are invoked automatically when you either open up a pentest results file on its own, or in the context of an FPR file. When you merge a pentest results file into an HP Fortify project, the pentest results will be persisted in the FPR file during a save.

Pentest issues have an analyzer attribute equal to pentest, and an analysis type attribute that reflects the tool or service (for instance, WebInspect issues have the WEBINSPECT analysis type. You can view these attributes through the standard Group By and Search mechanisms.

After you select a pentest issue, Audit Workbench displays the penetration test details on the Pentest Details tab.

About Public APIs

HP Fortify publishes public APIs so that you can create custom third-party parsers for pentest tools and services that are not included in HP Fortify's default distribution. The APIs are located in (fortify-public-*.jar), and can be used to compile your custom parser.

About Penetration Test Schema

HP Fortify also provides a generic penetration test schema (pentestimport.xsd) that you can view in Core/config/schemas. This provides another option for importing additional pentest results. Instead of creating a custom parser for your tool or service, you can translate the results into HP Fortify's generic format (using XSLT or a similar technology). You can then open or merge these translated results automatically, similar to the built-in parsers.
Chapter 4: Auditing Analysis Results

The following topics provide information about how to audit scan results uploaded to Audit Workbench.

Evaluating Issues

To evaluate and assign auditing values to an issue or group of issues:

1. In the analysis results panel, select the issue or group of issues.
   For information about the Analysis Evidence panel, see About the Analysis Evidence Panel on page 42.
2. Read the abstract on the Issue Summary tab, which provides high-level information about the issue, such as which analyzer found the issue.
   For example, “Command Injection (Input Validation and Representation, data flow)” indicates that this issue, detected by the data flow analyzer, is a command injection issue in the Input Validation and Representation kingdom.
3. Click the More Information link or the Issue Details tab to get more details about the issue.
4. On the Summary tab, assign values to the issue to represent your evaluation.
   Default choices in the Analysis menu are:
   - Not an issue
   - Reliability issue
   - Unknown
   - Suspicious
   - Exploitable
5. (Optional) In the Comments field, type a comment about the issue and your evaluation.

Performing Quick Audits

As you audit issues in Audit Workbench, you can use a keyboard combination to assign an analysis value to multiple selected issues.

To assign an analysis value to multiple issues simultaneously:

1. In the Issues Panel, select the issues that are to be assigned the same analysis value.
2. Press CTRL+SHIFT+A (CMD+SHIFT+A on Macintosh systems).
3. Press one of the following number keys:
   - To assign Not an Issue, press 1
   - To assign Reliability Issue, press 2
   - To assign Bad Practice, press 3
   - To assign Suspicious, press 4
   - To assign Exploitable, press 5
   - To assign a custom analysis value configured for your organization, press the number that corresponds to its position in the Analysis list on the Summary tab.

Audit Workbench provides shortcuts for only the first ten values in the Analysis list. (To assign the tenth value in the list, you press CTRL+SHIFT+A, and then press 0). If there is no value in the list for the key you press, no analysis value is assigned.
Using Custom Tags for Quick Audits

Instead of using the Analysis attribute for quick audits, you can use a custom tag that has been created for your organization.

To use a custom tag for quick audits:

1. Select **Options → Options**.
   The Options dialog box opens.
2. In the left panel, select **Interface Preferences**.
3. From the **Attribute Used for Quick Audit Action** list in the right panel, select a custom tag.
4. Click **OK**.

The keyboard shortcut functions just as it does for the Analysis attribute values. Audit Workbench provides shortcuts for only the first ten values in the list of custom tag values. (To assign the tenth value in the list, you press CTRL+SHIFT+A, and then press 0). If there is no value in the list for the key you press, no value is assigned.

For information about custom tags, see **About Custom Tags** on page 51. For instructions on how to create custom tags, see the *HP Fortify Software Security Center User Guide*.

Adding Screen Captures to Issues

You can attach a screenshot or other image to an issue. Attached images are stored in the FPR file and can be accessed from Software Security Center. Audit Workbench supports the following image formats:

- gif
- jpg
- png

To add an image to an issue

1. Select the issue.
2. In the Issue Auditing panel, click the **Screenshots** tab.
3. Click **Add**.
   The New Screen Shot dialog box opens.
4. Browse to and select the image file.
5. (Optional) In the **Description** box, type a description.
6. Click **Add**.

Viewing Images

After you add an image to an issue, the image is displayed on the right side of the **Screenshots** tab.

To view a full-size version and complete description of an image added to an issue:

1. In the Issue Auditing Panel, click the **Screenshots** tab.
2. In the list, click an image to view.
3. Click **Preview**.
Creating Issues for Undetected Vulnerabilities

You can add undetected vulnerabilities that you want to identify as issues to the issues list in the analysis results panel. You can audit such issues from the Summary tab, just as you do other issues.

To create an issue:
1. In the Source Code Viewer panel, select the object in the line of code.
2. Right-click the line that contains the issue, and then select Create New Issue from the shortcut menu.
   The Create New Issue dialog box opens.
3. Select the category to which the issue belongs, and then click OK.

In the analysis results panel, the issues list displays the file name and source code line number for the new issue next to a blue icon.

Suppressing Issues

As you assess successive scans of a project version, you might want to completely suppress some exposed issues. In Audit Workbench, it is useful to mark an issue as suppressed if you are sure that the specific vulnerability is not, and will never be, an issue of concern.

You might also want to suppress warnings for specific types of issues that might not be high priority or of immediate concern. For example, you can suppress issues that are fixed, or issues that you plan not to fix.

To suppress an issue, do one of the following:
• In the issues panel, select the issue, and then, on the Summary tab in the Issue Auditing panel, click the Suppress icon.
• Right-click the issue in the issues panel, and then select Suppress Issue.

Suppression marks the issue and all future discoveries of this issue as suppressed. As such, it is a semi-permanent marking of a vulnerability.

To display issues that have been suppressed, from the menu bar, select Options Show → Suppressed Issues.

To unsuppress an issue, do one of the following:
• In the issues panel, select the suppressed issue, and then, on the Summary tab in the Issue Auditing panel, click the Unsuppress icon.
• Right-click the issue in the issues panel, and then select Unsuppress Issue.
Submitting Issues as Bugs

You can submit issues to your bug-tracking application if integration between the applications has been configured.

To submit an issue as a bug:

1. Select the issue in the issues panel, then click File Bug icon on the Summary tab.
   If you are submitting a bug from Audit Workbench for the first time, the Configure Bugtracker Integration dialog box opens.
2. Select the bug-tracking application, and then click OK.
   The File Bug dialog box opens.
3. Review the issue description and change any values, if necessary.
4. Click Submit.

If your bug-tracking system requires you to log on, you must do so before you can file a bug through that user interface. The issue is submitted as a bug in the bug-tracking application.

Submitting an Issue as a Bug in Collaboration Mode

If you use HP Fortify Software Security Center, you can submit an issue as a bug from within Audit Workbench using a bug tracking system configured in HP Fortify Software Security Center. HP Fortify Software Security Center supports the following bug tracking systems:

- Bugzilla
- ALM
- HP Quality Center (on Windows systems only)

**Note:** You must be licensed to use the Collaboration Module feature in HP Fortify Software Security Center. For more information on using the Collaboration Module, see the HP Fortify Software Security Center User Guide.

To submit an issue as a bug:

1. Select the issue in the issues panel, and then, on the Summary tab, click File Bug.
   If this is the first time you are submitting a bug, the Configure Bugtracker Integration dialog box opens.
2. Select a bug-tracking application, and then click OK.
   The File Bug dialog box opens.
3. Specify the values if changes are needed and review the issue description. Depending on the integration and your bug-tracking application, the values include items such as the bug-tracking application URL, product name, severity level, summary, and version.
4. Click Submit.
5. If the bug-tracking system you selected requires that you log on, do so.
About Hotspot Ranking of Unaudited Issues

You can use Audit Workbench to generate hotspot values for issues that have been uncovered, but have not been audited. The hotspot values are similar to confidence values and are based on user-supplied audit data, and imported tracer data.

Hotspot ranking identifies unaudited issues that are similar to known vulnerabilities. For instance, if a user marks a dataflow issue as Exploitable, Audit Workbench interprets this as an indication that each node or element in the dataflow trace is valid. Unaudited issues that share some of the nodes are then treated as more likely to be exploitable. Hotspot ranking helps an auditor quickly identify the most important issues to review next.

Hotspot ranking uses the following information to calculate the likelihood that an unaudited issue is vulnerable to exploitation:

- Vulnerabilities identified by HP Fortify Runtime Application Protection, HP Fortify Program Trace Analyzer, and Fortify Source Code Analyzer
- Elements and characteristics shared with audited issues identified as vulnerabilities

*Table 8* provides descriptions of the hotspot rankings.

**Table 8: Hotspot Rankings**

<table>
<thead>
<tr>
<th>Hotspot Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likely</td>
<td>Shares a high number of elements or characteristics with issues identified by Runtime Application Protection or auditing as vulnerabilities</td>
</tr>
<tr>
<td>Possible</td>
<td>Shares some elements or characteristics with issues identified by Runtime Application Protection or auditing as vulnerabilities</td>
</tr>
<tr>
<td>Unknown</td>
<td>Does not contain enough evidence to calculate a hotspot ranking</td>
</tr>
<tr>
<td>Unlikely</td>
<td>Contains elements or characteristics also contained in audited issues marked <em>Not an Issue</em></td>
</tr>
<tr>
<td>Evidence</td>
<td>An issue previously identified by auditing or runtime analysis as a vulnerability</td>
</tr>
<tr>
<td></td>
<td>Audit Workbench uses the characteristics of evidence issues to calculate the hotspot ranking of unaudited issues.</td>
</tr>
</tbody>
</table>

Listing Issues by Hotspot Ranking

To filter issues by hotspot ranking:

1. Use Audit Workbench to audit one or more issues.
2. Select **Tools → Calculate Hotspot Ranking**.
   - Audit Workbench uses the characteristics of the audited issues to calculate the hotspot ranking for the unaudited issues.
3. From the **Filter Set** list, select **Hotspot**.
   - Audit Workbench displays the **Hotspot category** icons.
4. To view the issues in a given hotspot category, click the corresponding **Hotspot category** icon.
About Correlation Justification

A correlation occurs when an issue uncovered by one analyzer (HP Fortify SecurityScope, HP Fortify Static Code Analyzer, or HP WebInspect) is related directly or indirectly to an issue uncovered by another analyzer.

Correlated events help you identify issues that have a higher probability of being exploited. A vulnerability that is linked to other vulnerabilities might represent an issue that has multiple points of entry. For example, if HP WebInspect scan results are correlated with SCA scan results, this increases the likelihood that the associated SCA issues might be exploitable.

Audit Workbench provides additional information to help you resolve these correlated issues and mitigate the risks they present. In Audit Workbench, this additional information is presented as Correlation Justification.

For more information about issue correlation, see the *HP Fortify Runtime Hybrid Analysis User Guide*.

Using Correlation Justification

To use correlation justification:

1. In the issues panel, select a correlated issue. As illustrated below, a correlated issue is identified by a blue sphere in the lower left corner of the issues icon.

2. In the Issues Auditing panel, click the Correlated Issues tab.
The Issues Auditing panel lists the other issues that are correlated with the issue you first selected.

Because you first selected a correlated issue, the View Correlations button is available.

3. Click View Correlations.

The Correlation Justification dialog box opens and displays the following three panels:

- The correlated issues tree on the left displays all correlated issues within a correlated group, sorted based on analyzers.
- The relationship panel at the top right displays the correlation chain between issues. The chain describes any indirect or direct relationships between the two selected issues.
- The panel at the bottom right describes each correlation rule in the correlation chain displayed in the relationship panel.

4. To select two issues, press the CTRL key, and then click each issue.
The relationship panel displays the two issues and their relationships.

5. To inspect the attributes that correlate these two issues, move your cursor to each link in the relationship panel.

6. Click **OK**

Use correlation justification to gain insight into code vulnerabilities and understand why certain issues are correlated. This can help to reduce the time it takes to remediate the issues.
Chapter 5: Audit Workbench Reports

Audit Workbench provides a flexible reporting infrastructure based on user-configurable report templates. Report templates provide a number of optional sections and subsections that gather and present specific types of data. This chapter provides information about the default Audit Workbench reports and report templates, as well as instructions on how you can modify existing reports and create your own reports.

About Default Report Templates

HP Fortify supplies several pre-configured report templates, which you can either modify to suit your needs, or use as they come “out of the box.” Each report template includes several sections and subsections. The subsections provide charting and other data collection and presentation options.

About the Fortify Security Report

The Fortify Security Report is a high-level report that includes comprehensive analysis information and high-level details of the corresponding audit. This report also includes a high-level description and examples of the categories that have the highest priority.

Table 9 lists Fortify Security Report sections and their corresponding subsections.

Table 9: Fortify Security Report Sections and Subsections

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Summary</td>
<td>Issues Overview</td>
</tr>
<tr>
<td>Presents an overview of the scan. This includes an overview of issues, an overview of issues by Fortify Priority Order, and recommendations for issue remediation. This section is designed for management and project managers.</td>
<td>Editable overview of the issues, including the date of the scan, number of issues, name of the project, scan summary and total number of findings.</td>
</tr>
<tr>
<td>Issue Overview by Fortify Priority Order</td>
<td>Issues are categorized into the following four risk quadrants based on whether they have a high or low impact, and high or low likelihood of being exploited:</td>
</tr>
<tr>
<td>- Critical - High impact and high likelihood. Critical issues are easy for the attacker to discover and exploit to result in extensive asset damage.</td>
<td></td>
</tr>
<tr>
<td>- High - High impact but low likelihood. High priority issues are often difficult to discover and exploit, but can result in extensive asset damage.</td>
<td></td>
</tr>
<tr>
<td>- Medium - Low impact but high likelihood. Medium priority issues are easy to discover and exploit, but often result in little asset damage.</td>
<td></td>
</tr>
<tr>
<td>- Low - Low impact and low likelihood. Low priority issues can be difficult to discover and exploit and typically result in little asset damage.</td>
<td></td>
</tr>
<tr>
<td>You can present this information in a table or in a pie or a bar chart.</td>
<td></td>
</tr>
<tr>
<td>Recommendations and Conclusions</td>
<td>High-level recommendation on remediating the issues listed in the Issues Summary subsection.</td>
</tr>
</tbody>
</table>
### Table 9: Fortify Security Report Sections and Subsections

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Project Summary</strong>&lt;br&gt;Provides project summary information such as the code base, scan information, results certifications, and so on.</td>
<td><strong>Code-Based Summary</strong>&lt;br&gt;Summary of the codebase. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Scan Information</strong>&lt;br&gt;Analysis details. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Results Certification</strong>&lt;br&gt;Results certifications summary. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Attack Surface</strong>&lt;br&gt;Attack surface summary. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Filter Set Summary</strong>&lt;br&gt;Summary of the filter set used in the report. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Audit Guide Summary</strong>&lt;br&gt;Summary of the audit guide. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td><strong>Results Outline</strong>&lt;br&gt;Provides an outline of the results that SCA produced during the scan</td>
<td><strong>Overall Number of Results</strong>&lt;br&gt;Total number of results that SCA produced during the scan. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Vulnerability Examples by Category</strong>&lt;br&gt;Results summary of highest-level issues by category.</td>
</tr>
<tr>
<td><strong>Detailed Project Summary</strong>&lt;br&gt;Provides a detailed project summary.</td>
<td><strong>Files Scanned</strong>&lt;br&gt;List of all scanned files. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Reference Elements</strong>&lt;br&gt;List of all libraries that SCA used during the translation phase of analysis. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Rulepacks</strong>&lt;br&gt;List of Rulepacks that SCA used in the analysis. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td></td>
<td><strong>Properties</strong>&lt;br&gt;List of properties that SCA set during the analysis phase. You can edit the text element of this subsection.</td>
</tr>
<tr>
<td><strong>Issue Count by Category</strong>&lt;br&gt;Provides a chart of issues by category. This chart is user-configurable. You can choose to present the information in a table or as a pie or bar chart</td>
<td>N/A</td>
</tr>
</tbody>
</table>
About the Fortify Developer Workbook Report

The Fortify Developer Workbook report provides a high-level summary of the vulnerabilities detected during a scan. This includes a report summary and an issue summary by Fortify Priority Order. This report is designed for developers.

Table 10 lists Fortify Developer Workbook report sections and their corresponding subsections.

Table 10: Fortify Developer Workbook Report Sections and Subsections

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Overview</td>
<td></td>
</tr>
<tr>
<td>Provides a high-level overview of report findings.</td>
<td></td>
</tr>
<tr>
<td>Report Summary</td>
<td></td>
</tr>
<tr>
<td>Editable overview of vulnerability. This includes the date of the scan, the project name, and the total number of variabilities.</td>
<td></td>
</tr>
<tr>
<td>Issue Overview by Category</td>
<td></td>
</tr>
<tr>
<td>Chart of issues by category. You can present the information in a table or as a pie or a bar chart.</td>
<td></td>
</tr>
<tr>
<td>Issue Summary</td>
<td></td>
</tr>
<tr>
<td>Provides the number and categories of vulnerabilities.</td>
<td></td>
</tr>
<tr>
<td>Overall Number of list:suppressed</td>
<td></td>
</tr>
<tr>
<td>Total number of vulnerabilities. You can edit the text element of this subsection.</td>
<td></td>
</tr>
<tr>
<td>Results Outline</td>
<td></td>
</tr>
<tr>
<td>Provides an outline of the results that SCA produced during the scan.</td>
<td></td>
</tr>
<tr>
<td>Vulnerability Examples by Category</td>
<td></td>
</tr>
<tr>
<td>Results summary of highest-level issues by category.</td>
<td></td>
</tr>
</tbody>
</table>
### About the OWASP Top Ten 2004, 2007, 2010 Reports

The OWASP top ten reports provide high-level summaries of uncovered vulnerabilities organized based on the top ten issues identified by the Open Web Security Project (OWASP) for years 2004, 2007, and 2010. These reports include the sections and subsections described in Table 11.

**Table 11: OWASP Top Ten Report Sections and Subsections**

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report Overview</strong></td>
<td></td>
</tr>
<tr>
<td>Provides a high-level overview of report findings</td>
<td>Report Summary</td>
</tr>
<tr>
<td></td>
<td>Editable overview of vulnerabilities, including the date of the scan, the project name, and the total number of vulnerabilities.</td>
</tr>
<tr>
<td><strong>Issue Overview by Category</strong></td>
<td>Issue Overview by Category</td>
</tr>
<tr>
<td></td>
<td>Chart of issues grouped by a selected attribute such as category, kingdom, or analysis type. You can format information in a table or as a pie chart or bar chart.</td>
</tr>
<tr>
<td><strong>Issue Breakdown by OWASP Top Ten</strong></td>
<td></td>
</tr>
<tr>
<td>Provides a chart of issues organized by OWASP top ten security risks. You can present the information in a table or as a pie or bar chart.</td>
<td>N/A</td>
</tr>
<tr>
<td><strong>Results Outline</strong></td>
<td></td>
</tr>
<tr>
<td>Provides an outline of the results that HP Fortify SCA produced during the scan.</td>
<td>Vulnerability Examples by OWASP Top Ten</td>
</tr>
<tr>
<td></td>
<td>Lists the vulnerabilities organized by the OWASP top ten. You can use select listing to further refine and organize the vulnerabilities that Audit Workbench provides in the report.</td>
</tr>
</tbody>
</table>
About the Fortify Scan Summary Report

The Fortify scan summary report type provides high-level information based on the category of issues that SCA found as well as a project summary and a detailed project summary. *Table 12* provides descriptions of the report sections and subsections.

**Table 12: Fortify Scan Summary Report Sections and Subsections**

<table>
<thead>
<tr>
<th>Section</th>
<th>Subsection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Issue Count by Category</td>
<td>N/A</td>
</tr>
<tr>
<td>Provides a chart of issues by category; you can present the information in a table or as a pie or a bar chart.</td>
<td></td>
</tr>
<tr>
<td>Project Summary</td>
<td>Code Base Summary</td>
</tr>
<tr>
<td>Provides project summary information, including code base summary and general scan information.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Scan Information</td>
</tr>
<tr>
<td></td>
<td>Results Certification</td>
</tr>
<tr>
<td></td>
<td>Detailed Project Summary</td>
</tr>
<tr>
<td></td>
<td>Files Scanned</td>
</tr>
<tr>
<td></td>
<td>Reference Elements</td>
</tr>
<tr>
<td></td>
<td>Rulepacks</td>
</tr>
<tr>
<td></td>
<td>Properties</td>
</tr>
<tr>
<td></td>
<td>CommandLine Arguments</td>
</tr>
<tr>
<td></td>
<td>Warnings</td>
</tr>
</tbody>
</table>

- **Code Base Summary**
  Summary of the codebase that SCA scanned, including the location of the code, the number of files, lines of code, and the build label. You can edit the text element of this subsection.

- **Scan Information**
  Scan information, including the SCA version, machine name, and the name of the user who ran the scan. You can edit the text element of this subsection.

- **Results Certification**
  Results certifications information, including the results certification summary and the details of the results certification. You can edit the text element of this subsection.

- **Files Scanned**
  Lists all files SCA scanned. You can edit the text element of this subsection.

- **Reference Elements**
  List of libraries that SCA used during the translation phase. You can edit the text element of this subsection.

- **Rulepacks**
  List of Rulepacks that SCA used during the analysis. You can edit the text element of this subsection.

- **Properties**
  Lists the properties that SCA set during the analysis. You can edit the text element of this subsection.

- **CommandLine Arguments**
  Lists the arguments that the program passed to SCA during analysis. You can edit the text element of this subsection.

- **Warnings**
  Lists the warnings that occurred during analysis. You can edit the text element of this subsection.
About Modifying Report Templates

You can modify report templates from the Audit Workbench user interface, or by editing templates directly in the XML files that contain them. The following sections provide information about how to view report templates and customize them to address your reporting needs.

Viewing Report Templates

To open a report template:

1. On the Audit Workbench toolbar, click Reports.
   The Generate Report dialog box opens.

2. From the Report list, select a report template.
   The Generate Report dialog box displays the template settings for the selected report. The panel on the left lists the section titles included in the report. A report template typically includes several sections, which in turn, have subsections. You can select the sections to include in, or exclude from, a report. You can also edit the content displayed in each section.

3. To view detailed information about what a section contains, click the listed section title.
   On the right, the Generate Report dialog box displays the subsection headings and descriptions.
Selecting Report Sections

Each report template might have a few or several sections, which typically have subsections. You can select the sections to include in, or exclude from, a report. You can also edit the content displayed in each section.

To specify the sections to include in a report:

1. In the panel on the left side of the Generate Report dialog box, select the check boxes for the report section titles to include in the report, and clear the check boxes for the report section titles to exclude from the report.

   On the right, the Generate Report dialog box displays the subsection headings and descriptions for the report section selected in the panel on the left.

2. To exclude a subsection from the selected report section, clear the check box next to the subsection heading.

About Editing Report Subsections

Report subsections can include text, results lists, or charts. You can edit the subsection contents to display in a report. You can edit text, add or change text variables, or configure the issues shown in a chart or results list.

Editing Text Subsections in Reports

To edit a text subsection:

1. Select the check box next to the subsection title to include this text in the report.

   Audit Workbench displays description of the text below the subsection title.

2. Click Edit Text.

   A text box displays the text and variables to be included in the report.

3. Edit the text and text variables, as necessary.

As you edit text subsections, you can insert variables that are defined when you run the report. The following table lists the variables you can include.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$AUDIT_GUIDE_SUMMARY$</td>
<td>List filters created by answering Audit Guide questions</td>
</tr>
<tr>
<td>$CLASSPATH_LISTING$</td>
<td>JAR files used during scan, one relative path per line</td>
</tr>
<tr>
<td>$COMMANDLINE_ARGS$</td>
<td>Complete listing of command line arguments (same format as project summary)</td>
</tr>
<tr>
<td>$FILE_LISTING$</td>
<td>List of files scanned, each file in format &lt;relative file path&gt; # Lines # kb &lt;timestamp&gt;</td>
</tr>
<tr>
<td>$FILTERSET_DETAILS$</td>
<td>List of filters in use by current filter set</td>
</tr>
<tr>
<td>$FILTERSET_NAME$</td>
<td>Name of current filter set</td>
</tr>
<tr>
<td>$FORTIFY_SCA_VERSION$</td>
<td>SCA version</td>
</tr>
<tr>
<td>$LIBDIR_LISTING$</td>
<td>Libdirs specified during scan, one relative path per line</td>
</tr>
<tr>
<td>$LOC$</td>
<td>Total lines of code (executable lines of code)</td>
</tr>
<tr>
<td>$NUMBER_OF_FILES$</td>
<td>Total number of files scanned</td>
</tr>
<tr>
<td>$PROJECT_BUILD_LABEL$</td>
<td>Build label of project</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$PROJECT_LABEL$</td>
<td>Audit Workbench displays build-label when each FPR file that SCA generated passes build project with -build-project</td>
</tr>
<tr>
<td>$PROJECT_NAME$</td>
<td>Build ID</td>
</tr>
<tr>
<td>$PROPERTIES$</td>
<td>Complete list of properties set during analysis phase (same format as project summary)</td>
</tr>
<tr>
<td>$RESULTS_CERTIFICATIONS$</td>
<td>Complete certification detail with listing of validity on a per file basis (see project summary)</td>
</tr>
<tr>
<td>$RESULTS_CERTIFICATION_SUMMARY$</td>
<td>Short sentence describing certification (same format as project summary)</td>
</tr>
<tr>
<td>$RULEPACKS$</td>
<td>Complete list of Rulepacks used during analysis (same format as project summary)</td>
</tr>
<tr>
<td>$RUN_INFO$</td>
<td>Content from the Project Summary Runtime Information tab</td>
</tr>
<tr>
<td>$SCAN_COMPUTER_ID$</td>
<td>Hostname of machine on which scan was performed</td>
</tr>
<tr>
<td>$SCAN_DATE$</td>
<td>Date of analysis with the default formatting style for the locale</td>
</tr>
<tr>
<td>$SCAN_SUMMARY$</td>
<td>Summary of codebase scanned in the format # files, # lines of code</td>
</tr>
<tr>
<td>$SCAN_TIME$</td>
<td>Scan duration in minutes and seconds (mm:ss)</td>
</tr>
<tr>
<td>$SCAN_USER$</td>
<td>Username for the user who performed the scan</td>
</tr>
<tr>
<td>$SOURCE_BASE_PATH$</td>
<td>Source base path of the codebase</td>
</tr>
<tr>
<td>$TOTAL_FINDINGS$</td>
<td>Total number of findings, not including suppressed or removed issues</td>
</tr>
<tr>
<td>$VERSION_LABEL$</td>
<td>Audit Workbench displays build-label when each FPR file that SCA generated passes build version with build-version</td>
</tr>
<tr>
<td>$WARNINGS$</td>
<td>Complete listing of warnings that occurred (same format as project summary)</td>
</tr>
<tr>
<td>$WARNING_SUMMARY$</td>
<td>Count of warnings found in scan</td>
</tr>
</tbody>
</table>
Editing Result List Subsections in Reports

To edit a result list subsection:

1. Select the check box next to the subsection title to include this text in the report. A description of the results list displays below the subsection title.
2. Click the issues **Listing** heading to reveal the configuration options for the subsection.
3. From the grouping list, select the attributes used to group the results. (For information about groupings, see *About Grouping Issues* on page 35.)

   **Note:** If you list issues based on the Category grouping, the recommendations, abstract, and category explanation are included in the report.
4. To filter the reported results further, do one of the following:
   - Type a query in the **Refine Issues in Subsection** box
   - Click the **Advanced** search link, and use the advanced search wizard to compose a query. (For information about search syntax, see *About Searching Issues* on page 45 and *Performing Advanced Searches* on page 50.)

   **Note:** To see the **Advanced** search link, you might need to resize (widen) the dialog box.

   The **Refine Issues in Subsection** box displays the query.
5. Select or clear the **Limit number of Issues in each group** check box.
6. If you select the check box, then type the number of issues to display per group.
7. To include information about an external category for each issue reported, select the **Include External Category for Each Issue** check box.

Editing Chart Subsections in Reports

To edit a chart subsection:

1. From the list of attributes, select the attribute to use to group issues in the chart. (For information about grouping issues, see *About Searching Issues* on page 45.)
2. To filter the reported results further, do one of the following:
   - Type a query in the **Refine Issues in Subsection** box
   - Click the **Advanced** search link, and use the advanced search wizard to compose a query. (For information about search syntax, see *About Searching Issues* on page 45 and *Performing Advanced Searches* on page 50.)

   **Note:** To see the **Advanced** search link, you might need to resize (widen) the dialog box.

   The **Refine Issues in Subsection** box displays the query.
3. To specify the chart format, select **Table**, **Pie**, or **Bar**.
Saving Modified Report Template Settings

You can save your changes to a report template either as the default settings for the template you changed, or as a new template.

To save modified settings as the default report template settings:

- In the Generate Reports dialog box, after you modify the report settings, click **Save as Default**.

When you next select the report template name from the Report list, the settings displayed in the Generate Report dialog box reflect your changes.

To save modified settings as a new report template:

1. In the Generate Reports dialog box, after you modify the report settings, click **Save as New Template**.
   - The Enter Value dialog box opens.
2. In the text box, type a name for the new report template.
3. Click **OK**.

When you next open Generate Report dialog box, the Report list includes the new template name.
About Report Template XML Files

Audit Workbench saves report templates as XML files. Although you can use the Audit Workbench graphical interface to modify report templates and create custom report templates, you can also edit the XML directly to modify and create report templates.

The default location for the report template XML files on the Windows platform is:

Documents and Settings\<user>\Local Settings\Application Data\Fortify\config\AWB-\<version>\reports

where <user> is the name of the person who is logged on and using Audit Workbench and <version> is the version of Audit Workbench you are using.

On other platforms, the path looks similar to the following:

<user_directory>/..fortify/config/AWB-<version>/reports

You can also customize the logos used in the reports by specifying paths or replacing header.png and footer.png in this directory.

Adding Report Sections in the XML Files

You can add report sections by editing the XML files. In the structure of the XML, the ReportSection tag defines a new section. It includes a Title tag for the section name, and it must include at least one Subsection tag to define the contents of the section in the report.

The following XML is the Results Outline section of the HP Fortify Security Report:

```xml
<ReportSection enabled="false" optionalSubsections="true">
  <Title>Results Outline</Title>
  <SubSection enabled="true">
    <Title>Overall number of results</Title>
    <Description>Results count</Description>
    <Text>The scan found $TOTAL_FINDINGS$ issues.</Text>
  </SubSection>
  <SubSection enabled="true">
    <Title>Vulnerability Examples by Category</Title>
    <Description>Results summary of the highest severity issues. Vulnerability examples are provided by category.</Description>
    <IssueListing limit="1" listing="true">
      <Refinement>severity:(3.0,5.0] confidence:[4.0,5.0]</Refinement>
      <Chart chartType="list">
        <Axis>Category</Axis>
        ...
      </Chart>
    </IssueListing>
  </SubSection>
</ReportSection>
```

In this example, the Results Outline section contains two subsections. The first subsection is a text subsection named Overall number of results. The section subsection is a results list named Vulnerability Examples by Category. A section can contain any combination of subsections as its contents.
About Adding Report Subsections in the XML Files

In the report sections, you can add subsections or edit the contents of the subsections. Subsections can generate text, results lists, or charts.

Adding Text Subsections

In a text subsection, you can include the Title tag, the Description tag, and the Text tag. In the Text tag, you can provide the default content although the user can edit the content before generating a report. For a description of the text variables available to use in text subsections, see About Editing Report Subsections on page 78. The following XML is the Overall number of results subsection in the Results Outline section:

```
<SubSection enabled="true">
  <Title>Overall number of results</Title>
  <Description>Results count</Description>
  <Text>The scan found $TOTAL_FINDINGS$ issues.</Text>
</SubSection>
```

In this example, the text subsection is titled Overall number of results. The description text to describe the purpose of the text is Results count. The text in the text field that the user can edit before running a report uses one variable named $TOTAL_FINDINGS$.

Adding Results List Subsections

In a results list subsection, you can include the Title tag, the Description tag, and the IssueListing tag. In the IssueListing tag, you can define the default content for the limit and set listing to true. You can include the Refinement tag either with or without a default statement although the user can edit the content before generating a report. To generate a results list, the Chart tag attribute chartType is set to list. You can also define the Axis tag. The following XML is the Vulnerabilities Examples by Category subsection in the Results Outline section:

```
<SubSection enabled="true">
  <Title>Vulnerability Examples by Category</Title>
  <Description>Results summary of the highest severity issues. Vulnerability examples are provided by category.</Description>
  <IssueListing limit="1" listing="true">
    <Refinement>severity:(3.0,5.0] confidence:[4.0,5.0]</Refinement>
    <Chart chartType="list">
      <Axis>Category</Axis>
    </Chart>
  </IssueListing>
</SubSection>
```

In this example, the results list subsection is titled Vulnerability Examples by Category. The description text to describe the purpose of the subsection is Results summary of the highest severity issues. Vulnerability examples are provided by category. This subsection list (listing=true) one issue (limit="1") per category (the value of the Axis tag) where issues match the statement severity:(3.0,5.0] confidence:[4.0,5.0] (the value of the Refinement tag).
Adding Charts Subsections

In a chart subsection, you can include the Title tag, the Description tag, and the IssueListing tag. In the IssueListing tag, you can define the default content for the limit and set listing to false. You can include the Refinement tag either with or without a default statement although the content can be edited by the user before generating a report. To generate a pie chart, the Chart tag attribute chartType is set to pie. The options are table, pie, and bar. This is a setting that the user can change before generating the report. You can also define the Axis tag.

The following code shows an example of a charts subsection:

```xml
<SubSection enabled="true">
  <Title>New Issues</Title>
  <Description>A list of issues discovered since the previous analysis</Description>
  <Text>The following issues have been discovered since the last scan:</Text>
  <IssueListing limit="-1" listing="false">
    <Refinement />
    <Chart chartType="pie">
      <Axis>New Issue</Axis>
    </Chart>
  </IssueListing>
</SubSection>
```

In the subsection above, a chart (limit="-1" listing="false") will have the title New Issues and a text section containing The following issues have been discovered since the last scan. This chart will include all issues since the Refinement tag is empty and will group the issues on the value of the New Issue (the value of the Axis tag). This chart will be displayed as a pie chart (chartType="pie").
Generating Reports

Use this section to learn how to generate reports, create a new report template, or make changes to a default report template.

**Note:** Although the procedure described here uses a Fortify report template, you can follow these steps using a report template that you have customized and saved.

To generate a report based, create a new report template, or make changes to a default report template:

1. Select **Tools → Generate Report**.
   
The Generate Report dialog box opens.

2. From the **Report** list, select the report to generate.
   
The left panel lists the sections that you can include in (or exclude from) the selected report.

3. To see the subsection headings included in a report section, in the left panel, click the section title.
   
The subsections are listed and described on the right. For example, the Executive Summary section of the Fortify Security Report can include the Issues Overview, Issue Summary by Fortify Priority Order, and Recommendations and Conclusions subsections.

4. In the left panel, select or clear the check boxes for the sections to include in (or exclude from) the selected report.

5. Select a section, and then, on the right, configure its subsections for the report.

6. Repeat **Step 5** for each section to be included in the report.

7. To control whether the report displays suppressed, hidden, and removed issues:
   
   a. At the top right, click **Visibility Settings**.

   ![Generate Report dialog box with Visibility Settings option highlighted]
b. In the Report Issue Preferences dialog box, select the check boxes for the suppressed, hidden, or removed to display in the report, and then click OK.

For descriptions of suppressed, hidden, and removed issues, see Viewing Suppressed, Removed, and Hidden Issues on page 32.

8. Click Save Report.

The Save Report dialog box opens.

9. Make any necessary changes to the report details, including its location and format, and then click Save.
Chapter 6: Functions View

About the Functions View

Static Code Analyzer identifies all functions declared or called in your source code. You can use the Functions view in Audit Workbench to determine where a function is located in the source code, whether the function was covered by a security rule, and which rule IDs matched the function. You can also list the functions that Static Code Analyzer identified as tainted source and view only the functions not covered by rules applied in the most recent scan.

Opening the Functions View

To open the Functions view:
1. In Audit Workbench, open the Audit window.
2. Select Options → Show View → Functions.

Audit Workbench displays the Functions view in the top right panel.

3. To view coverage information about top-level (global) functions, expand the Top-level functions node.
4. To view descriptions of the icons displayed to the left of each function, click the **Legend** link.

![Legend](image.png)

The following icons indicate the rule coverage status of the function:

- A solid red square indicates that the function is not identified by any rules.
- A blue triangle indicates that the function is in a package covered by the Secure Coding Rulepacks.
- A green circle indicates that the function matches a rule defined in either the Secure Coding Rulepacks or a custom Rulepack. (Matching a rule may not result in a reported issue.)

**Sorting and Viewing Functions**

To change the order of, or to hide or show functions:

1. Open the **Functions** view.
2. From the **Show** list, select one of the following:
   - To display all functions, select **All**.
   - To display functions not covered by rules, select **Not Covered by Rules**.
   - To display functions that are identified as a source of tainted data by the Secure Coding Rulepacks or a custom Rulepack used in the most recent scan, select **Taint Sources**.
3. From the **Group By** list, select one of the following sorting methods:
   - To sort functions based on package and class, select **Package**.
   - To sort listed functions by class, select **Class**.
   - To sort listed functions alphabetically, select **Function**.

Audit Workbench updates the **Functions** view.

**Locating Functions in Source Code**

You can list the file name and line number where the function appears in the source code from the **Functions** view.

To show where a function is located in code:

1. In the **Functions** view, right-click a function, and then select **Find Usages** on the shortcut menu.

   The **Search** view (at center bottom) lists the file locations and line numbers in which the function is used.

2. To jump to a line of code where the function is used, click the corresponding row in the **Search** view.

**Locating Classes in Source Code**

To see where classes are used in the source code:

1. In the **Functions** view, right-click a class, and then select **Find Usages** on the shortcut menu.

   The **Search** view (at center bottom) lists the file locations and line numbers in which the class is used.
2. To jump to a line of code where the class is used, click the corresponding row in the Search view.

For functions defined in the source code, you can open the declaration in the Source view by right-clicking on a function and selecting Open Declaration. The source code is displayed with the line highlighted. Alternatively, you can double-click functions to display the declaration.

**Determining Which Rules Matched a Function**

You can display the Rule ID for all the rules that matched a function. When rules match a function, a green circle icon displays next to it.

SCA can match a rule to functions without finding an issue related to the rule. For example, a tainted data source rule matches the source function but the tainted data entering at that function does not reach a sink.

**Note:** To use the rule ID to locate related issues, see About Searching Issues on page 45, or create visibility or folder filters.

To display the rule IDs:

1. Open a project in Audit Workbench.
2. Open the Functions view.
3. Right-click a function, and then select Show Matched Rules from the shortcut menu.

The Matched Rules dialog box displays a list of the rule IDs with the vulnerability category name (if applicable) and the Rulepack file name.

**Writing Rules for Functions**

You can launch the Custom Rules Wizard from the Functions view. The wizard is perpetuated with the language of the source code and the function is automatically identified.

To write a rule for a function:

1. Open a project in Audit Workbench.
2. Open the Functions view.
3. To create a rule:
   a. Right-click a function, and then select Generate Rule for Function from the shortcut menu.
   b. Select the rule that best matches the behavior or vulnerability category.
   c. Enter the information as directed by the wizard, and save the new rule to a custom Rulepack.
4. To reascend the translated files with the custom Rulepack:
   a. Click Scan.
   b. Click Configure Additional Options.
      The Additional Options dialog box opens.
   c. Click Add Additional Rulepack.
   d. Browse to and select the custom Rulepack, and then click Open.
   e. Make sure that the Rulepack is selected in the Custom Rulepack box, and then click OK.
   f. Click Run Scan.
      After the scan is completed, the project is updated.
5. Click OK
6. To verify that the rule matched the function:
• Right-click the function, and then select **Show Matched Rules** from the shortcut menu. The Matched Rules dialog box opens.
• Verify that at least one rule ID matches the ID of the rule you created.

The function is now covered by a custom Rulepack and is displayed with a green circle next to it.

## Creating Custom Cleanse Rules

You can create custom cleanse rules for specific functions from Audit Workbench.

To create a cleanse rule for a function:

1. Right-click the function, and then select **Write Rule for This Function** from the shortcut menu.

   The Custom Rules Wizard opens.

2. In the templates list, expand the **DataflowCleanseRule** folder, and then select **Generic Validation Rule**.

3. Click **Next**.

4. On the **Rule Language** step, select the source code language, and then click **Next**.

5. On the **Validation Function Information** step, enter the regular expressions for the package, class and function.

6. Verify that the information is correct, and then click **Next**.

7. On the **Function Argument** step, select the argument to cleanse, and then click **Next**.

   On the **Custom Rulepack Selection** step, select the Rulepack to which you want to add the rule, and then click **OK**.
Chapter 7: Troubleshooting

This chapter provides information on how to troubleshoot problems you might encounter working with Audit Workbench.

Topics include:

- Using the Debug Option
- Addressing the org.eclipse.swt.SWTError Error
- About Out of Memory Errors
- Determining the Amount of Memory Used by External Processes
- Resetting the Default Views

Using the Debug Option

If you are encountering errors, you can enable the debugging option to help troubleshoot. To set all the thick tools to debug mode, set the option listed in Step 2 to true.

To enable debugging in all thick tools:

1. Navigate to the `<install>`\Core\config directory and open the `fortify.properties` file.
2. Uncomment the following lines and leave the value of each option set to true:

   ```
   #com.fortify.Debug=true
   #to selectively place a thick tool into debug mode (unnecessary if com.fortify.Debug is set to true)
   #com.fortify.awb.Debug=true
   #com.fortify.eclipse.Debug=true
   #com.fortify.VS.Debug=true
   ```

   **Note:** The `com.fortify.Debug` option is not designed to put Source Code Analyzer into default mode.

For help diagnosing the problem, send the log files to HP Fortify Software support.

For HP Fortify errors, see:

- `<fortify_data_directory>/awb/log/awb.log`

For errors related to Eclipse, see:

- `<fortify_data_directory>/awb/.metadata/.log`

where:

- On Unix, `<fortify_data_directory>` is the directory `~/.fortify`
- On Windows, `<fortify_data_directory>` is the directory `Local Settings/Application Data/Fortify` in the user directory under Documents and Settings.

Addressing the org.eclipse.swt.SWTError Error

On Unix systems, Audit Workbench can fail to start, resulting in the following error:

```
org.eclipse.swt.SWTError: No more handles [gtk_init_check() failed]
```

If you see this error, check to make sure that X11 is configured correctly and that your `DISPLAY` variable is set.
About Out of Memory Errors

The following two scenarios might trigger out-of-memory errors in Audit Workbench:

**Scenario 1:** An uploaded FPR file is too large

To solve this problem, increase the amount of memory allocated for running Audit Workbench.

To increase the memory allocated for Audit Workbench, set the environment variable `AWB_VM_OPTS` to allocate memory for Audit Workbench. (For example, set `AWB_VM_OPTS=-Xmx700M` to allocate 700 MB to Audit Workbench.)

**Note:** If you choose to set `AWB_VM_OPTS`, do not allocate more memory than is physically available. Overallocation degrades performance.

**Scenario 2:** Running an SCA scan through the advanced scan wizard

In this scenario, the SCA instance of SCA executed during the scan is running out of memory. To resolve this problem, increase the amount of memory allocated to SCA.

Allocating More Memory for Static Code Analyzer

To increase the memory allocated for SCA, do one of the following:

- Type the memory settings directly into the text box that displays the translation and scan arguments. (This is the last step of the Scan Runner wizard.) This passes the memory allocation option to SCA. This is the direct method that does not require a restart.
- Set the environment variable `SCA_VM_OPTS` before your start Audit Workbench, and then start Audit Workbench. For example, to allocate 700 MB to SCA, set the variable to `-Xmx700M`.

**Note:** If you choose to set `SCA_VM_OPTS`, do not allocate more memory than is physically available. Overallocation degrades performance.

As a guideline, assuming no other memory intensive processes are running, do not allocate more than two thirds of the available memory on the system.

If you are using Macintosh OS X, edit the following file to modify the `-Xmx500m` argument to `-Xmx700m` or higher:

```
<install_directory>/Auditworkbench.app/contents/MacOS/eclipse.ini
```

In Audit Workbench, issue information is persisted to disk. This persisted information is reloaded on demand and thereby decreases the required memory footprint of Audit Workbench. To prevent out-of-memory errors, you may set a value in the `fortify.properties` file to take advantage of the information persisted to disk functionality. The value to set is as follows:

```
com.fortify.model.PersistDataToDisk = true
```

**Note:** You can allocate a maximum of approximately 1500MB for a 32-bit version of Audit Workbench (as imposed by the underlying JVM). However, this number can vary, depending on your operating system and hardware.

For more information about the advanced scan wizard, see *Scanning Complex Projects* on page 20.
Determining the Amount of Memory Used by External Processes

You can use the `com.fortify.model.ExecMemorySetting` setting in the `fortify.properties` file to determine how much memory external processes such as iidmigrator or the Event Bridge use. The default setting is as follows:

```
com.fortify.model.ExecMemorySetting = 600
```

The value for this setting, which is expressed in MB, is translated into maximum heap size. In this case, 600 equates to `-Xmx600M`.

Resetting the Default Views

If you have closed or moved panels, such as the issues panel or the Summary panel, you can reset the user interface to the default state.

To reset the user interface to the default state:

1. Select **Options → Options**.
2. Click **Interface Preferences**.
3. Click **Reset Interface**.
Appendix A: Sample Files

Your HP Fortify software installation includes a number of sample files that you can use when testing or learning to use SCA. The sample files are located in the following directory:

\( <HP\_Fortify\_Install>/Samples \)

The `Samples` directory contains two subdirectories: `basic` and `advanced`. Each code sample includes a `README.txt` file that provides instructions on how to scan the code in SCA and view the output in Audit Workbench.

The `basic` subdirectory includes an assortment of simple language-specific samples. The `Advanced` subdirectory contains more advanced code samples, including samples that enable you to integrate SCA with your bug-tracking system.

Basic Samples

*Table 13* lists the sample files in the `basic` subdirectory, a short description of each file, and the vulnerabilities identified. Each sample includes a `README.txt` file that provides details and instructions on its use.

<table>
<thead>
<tr>
<th>Sample File Folder</th>
<th>Contents</th>
<th>Vulnerabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpp</td>
<td>Includes a C++ sample file and instructions for testing a simple dataflow vulnerability. It requires a gcc or cl compiler.</td>
<td>Command Injection, Memory Leak</td>
</tr>
<tr>
<td>database</td>
<td>Includes a database.pks sample file. This SQL sample includes issues that can be found in SQL code.</td>
<td>Access Control: Database</td>
</tr>
<tr>
<td>eightball</td>
<td>Includes EightBall.java, a Java application that exhibits bad error handling. It requires an integer as an argument. If you supply a filename instead of an integer, it displays the file contents.</td>
<td>Path Manipulation, Unreleased Resource: Streams, J2EE Bad Practices: Leftover Debug Code</td>
</tr>
<tr>
<td>formatstring</td>
<td>Includes formatstring.c file. It requires a gcc or cl compiler.</td>
<td>Format String</td>
</tr>
<tr>
<td>javascript</td>
<td>Includes the sample.js JavaScript file.</td>
<td>Cross Site Scripting (XSS), Open Redirect</td>
</tr>
<tr>
<td>nullpointer</td>
<td>Includes the NullPointerSample.java file.</td>
<td>Null Dereference</td>
</tr>
<tr>
<td>php</td>
<td>Includes both sink.php and source.php files. Analyzing source.php surfaces simple dataflow vulnerabilities and a dangerous function.</td>
<td>Cross Site Scripting, SQL Injection</td>
</tr>
<tr>
<td>sampleOutput</td>
<td>Includes a sample output file (WebGoat5.0.fpr) from the WebGoat project located in the Samples/advanced/webgoat directory.</td>
<td>Example input for Audit Workbench.</td>
</tr>
</tbody>
</table>
Table 13: Basic Samples

<table>
<thead>
<tr>
<th>Sample File Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>stackbuffer</td>
<td>Includes stackbuffer.c. A gcc or cl compiler is required.</td>
</tr>
<tr>
<td>toctou</td>
<td>Includes toctou.c file.</td>
</tr>
<tr>
<td>vb6</td>
<td>Includes command-injection.bas file.</td>
</tr>
<tr>
<td>vbscript</td>
<td>Includes source.asp and sink.asp files.</td>
</tr>
<tr>
<td>configuration</td>
<td>This is a sample J2EE application that has vulnerabilities in its web module deployment descriptor -web.xml.</td>
</tr>
<tr>
<td>c++</td>
<td>Includes a sample Visual Studio 2005 solution: Sample.sln, Sample1.cpp, Sample.vcproj, stafx.cpp, stafx.h. You need to have Microsoft Visual Studio Visual C/C++ 2005 (or newer) installed. You should also have the Fortify Analyzers installed, with the plugin for the Visual Studio version you are using. The code includes a Command Injection issue and an Unchecked Return Value issue.</td>
</tr>
<tr>
<td>crosstier</td>
<td>This is a sample that has vulnerabilities spanning multiple application technologies (Java, PL/SQL, JSP, struts). The output should contain several issues of different types, including two Access Control vulnerabilities. One of these is a cross-tier result. It has a data flow trace from user input in Java code that can affect a SELECT statement in PL/SQL.</td>
</tr>
<tr>
<td>csharp</td>
<td>This is a simple C# program that has SQL injection vulnerabilities. Versions are included for VS2003, VS2005, and VS2010. Upon successful completion of the scan, you should see the SQL Injection vulnerabilities and one Unreleased Resource vulnerability. Other categories might also be present, depending on the Rulepacks used in the scan.</td>
</tr>
<tr>
<td>customrules</td>
<td>Several simple source code samples and Rulepack files that illustrate rules interpreted by four different analyzers: semantic, data flow, control flow, and configuration. This directory also includes several miscellaneous real-world rules samples that may be used for scanning real applications.</td>
</tr>
</tbody>
</table>

Advanced Samples

Table 14 provides a list of the sample files in the advanced subdirectory (<HP_Fortify_Install>\Samples\advanced). Each sample includes a README.txt file that provides further details and instructions on its use.

Table 14: Advanced Samples

<table>
<thead>
<tr>
<th>Sample File Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bugzilla</td>
<td>Includes a Build.xml file built using the Audit Workbench bugtracker plugin framework. The plugin includes the same functionality as the built-in Bugzilla plugin so that it can be used as a guide to creating your own plugin.</td>
</tr>
<tr>
<td>c++</td>
<td>Includes a sample Visual Studio 2005 solution: Sample.sln, Sample1.cpp, Sample.vcproj, stafx.cpp, stafx.h. You need to have Microsoft Visual Studio Visual C/C++ 2005 (or newer) installed. You should also have the Fortify Analyzers installed, with the plugin for the Visual Studio version you are using. The code includes a Command Injection issue and an Unchecked Return Value issue.</td>
</tr>
<tr>
<td>configuration</td>
<td>This is a sample J2EE application that has vulnerabilities in its web module deployment descriptor -web.xml.</td>
</tr>
<tr>
<td>crosstier</td>
<td>This is a sample that has vulnerabilities spanning multiple application technologies (Java, PL/SQL, JSP, struts). The output should contain several issues of different types, including two Access Control vulnerabilities. One of these is a cross-tier result. It has a data flow trace from user input in Java code that can affect a SELECT statement in PL/SQL.</td>
</tr>
<tr>
<td>csharp</td>
<td>This is a simple C# program that has SQL injection vulnerabilities. Versions are included for VS2003, VS2005, and VS2010. Upon successful completion of the scan, you should see the SQL Injection vulnerabilities and one Unreleased Resource vulnerability. Other categories might also be present, depending on the Rulepacks used in the scan.</td>
</tr>
<tr>
<td>customrules</td>
<td>Several simple source code samples and Rulepack files that illustrate rules interpreted by four different analyzers: semantic, data flow, control flow, and configuration. This directory also includes several miscellaneous real-world rules samples that may be used for scanning real applications.</td>
</tr>
</tbody>
</table>
### Table 14: Advanced Samples

<table>
<thead>
<tr>
<th>Sample File Folder</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ejb</strong></td>
<td>A sample J2EE cross-tier application with Servlets and EJBs.</td>
</tr>
<tr>
<td><strong>filters</strong></td>
<td>A sample that uses sourceanalyzer’s –filter option.</td>
</tr>
<tr>
<td><strong>findbugs</strong></td>
<td>A sample that demonstrates how to run FindBugs static analysis tool together with the Fortify Source Code Analysis Engine (Fortify SCA Engine) and filter out results that overlap.</td>
</tr>
<tr>
<td><strong>HPQC</strong></td>
<td>A sample that demonstrates the Audit Workbench bugtracker plugin framework by implementing a plugin to HP Quality Center. This plugin communicates with an HPQC server instance through the HPQC client-side addin. The bug tracker talks to the addin through a COM interface, and the addin handles the communication to the server.</td>
</tr>
<tr>
<td><strong>java1.5</strong></td>
<td>Includes ResourceInjection.java. The result file should have a Path Manipulation result and a J2EE Bad Practices result.</td>
</tr>
<tr>
<td><strong>javaAnnotations</strong></td>
<td>Includes a sample application that illustrates problems that may arise from its use and how to fix the problems using the Fortify Java Annotations. The goal of this example is to illustrate how the use of Fortify Annotations can result in increased accuracy in the reported vulnerabilities. The accompanying README file illustrate the potential problems and solutions associated with vulnerability results.</td>
</tr>
<tr>
<td><strong>JavaDoc</strong></td>
<td>JavaDoc directory for the bugtrackers, public-api, and WSClient.</td>
</tr>
<tr>
<td><strong>maven-plugin</strong></td>
<td>Tests can be run on any projects that use Maven (for instance those included in the samples directory, or WebGoat 5.3: <a href="http://code.google.com/p/webgoat/">http://code.google.com/p/webgoat/</a>)</td>
</tr>
<tr>
<td><strong>webgoat</strong></td>
<td>WebGoat test J2EE web application provided by the Open Web Application Security Project (<a href="http://www.owasp.org">http://www.owasp.org</a>). This directory contains the WebGoat 5.0 sources. WebGoat java sources can be used directly for java vulnerability scanning via Fortify Source Code Analysis Engine.</td>
</tr>
</tbody>
</table>
Appendix B: Static Analysis Results Prioritization

To provide the most benefit from static analysis with the least amount of time and effort, Audit Workbench prioritizes static analysis results so that you can focus on the most pressing problems first. The following topics describe how HP Fortify Static Code Analysis automatically prioritizes the scan results displayed in Audit Workbench.

About Results Prioritization

Static Code Analysis divides static analysis findings into four risk quadrants: critical, high, medium, and low. Membership in each quadrant depends upon whether the finding has a high or low impact and high or low likelihood of occurring.

Quantifying Risk

Now that we have explained how we use impact, likelihood, and remediation information, we will explain how we quantify these values as part of the static analysis process.

Since it is not possible to determine if or when an organization will suffer consequences related to a particular vulnerability, Static Code Analysis takes a probabilistic approach to prioritizing vulnerabilities. Risk is defined quantitatively, as follows:

$$\text{risk} = \text{impact} \times \text{likelihood}$$

The risk that a vulnerability poses is equal to the impact of the vulnerability multiplied by the likelihood that the impact will occur. We define impact as the negative outcome resulting from a vulnerability and likelihood as the probability that the impact will come to pass.

Impact can come in many forms. For example, an organization might lose money or reputation because of a successful attack, or it might lose business opportunity because the presence of a vulnerability causes a system to fail a regulatory compliance check.

Two factors contribute to the likelihood that a particular vulnerability will cause harm:

- The probability that the vulnerability will be discovered (by an attacker or an auditor)
- The conditional probability that, once found, the vulnerability will be exploited
These probabilities change as the computer security field advances. New vulnerability assessment techniques make it easier to find vulnerabilities, and new attack techniques increase the set of vulnerabilities that can be exploited. Progressively better vulnerability prevention, mitigation, and recovery strategies help counterbalance these advances.

For example, consider Race Condition: Singleton Member Field vulnerabilities, which occur when code assigns a value associated with a particular user session to a member variable of a singleton object in a web application. Since, under the singleton model, the same class instance is used to service all requests, values from one user session can spill over into another user’s session. The following code demonstrates a singleton member field race condition:

```java
public class GuestBook extends HttpServlet {
    String name, password;
    protected void doPost (HttpServletRequest request, HttpServletResponse response) {
        name = request.getParameter("username");
        password = request.getParameter("password");
        if (DBUtils.lookupUser(username, password)) {
            accessSensitiveResources();
        }
    }
}
```

Although this vulnerability is fairly simple to exploit after it is found, finding race conditions can be difficult because successful attacks often depend on very precise timing. Therefore, this class of vulnerability has a low likelihood, which primarily reflects the difficulty involved in finding the vulnerability.

For an example of a vulnerability whose likelihood is primarily governed by how difficult it is to exploit, consider HTTP Header Manipulation, which occurs when unvalidated user input is included in an HTTP response header and can enable cross-site scripting, HTTP response splitting, and cache poisoning, among other attacks. The following code demonstrates a header manipulation vulnerability.

```java
String author = request.getParameter(AUTHOR_PARAM);
Cookie cookie = new Cookie("author", author);
cookie.setMaxAge(cookieExpiration);
response.addCookie(cookie);
```

In this case identifying a vulnerable application is often quite simple because the vulnerability is evident in web traffic returned from the server. Crafting a meaningful exploit, however, typically involves a deep understanding of the application’s business logic, ready access to a pool of legitimate users, and in some cases, a working knowledge of the network topography between the server and the users. Therefore, this class of vulnerability has a low likelihood because it is difficult to exploit.

**Inside a Static Analysis Engine**

The first step in analyzing a code base is translation. In this step, Fortify SCA converts the application source code into an intermediate format appropriate for analysis (NST). Once this intermediate model is built, SCA is ready to analyze the program. The input to analysis is the intermediate model and a set of secure coding rules that describe relevant properties of APIs used by the source code. SCA then uses a combination of different analysis techniques to identify potential security issues. Finally, after analysis is complete, SCA produces a results file that contains the details of the analysis.

Once SCA produces a results file, automated processing and human review can convert issues into findings. Findings, which represent specific problems with the code base, sometimes map one-to-one with issues. However, in other cases, multiple related issues might be combined into a single finding. For example, every form that submits a request without including a unique token might produce an issue related to CrossSite Request Forgery (CSRF), but these issues are more useful when they are combine into a single finding that indicates the application as a whole is vulnerable to CSRF attacks.
On occasion the static analysis process goes wrong. Depending on the rules and the analysis algorithms in use, a static analysis engine can produce false positives (reported vulnerabilities where no vulnerabilities exist) or false negatives (unreported vulnerabilities) or both.

Estimating Impact and Likelihood with Input from Rules and Analysis

HP Fortify Static Code Analysis estimates the impact of a discovered vulnerability based on its type. The impact value is associated with the static analysis rule that defines the vulnerability. In this way, results can indicate that a category such as cross-site scripting has a higher impact than a category such as null pointer dereference.

To compute the likelihood portion of the risk equation, Static Code Analysis draws on values from the rules used for analysis, the analysis process itself, and from a human auditor (if an individual has reviewed the results.) The likelihood of a finding is computed by combining the accuracy of the rule and the confidence in the analysis with the probability that the vulnerability will be discovered and acted upon, as follows:

\[ \text{likelihood} = \text{accuracy} \times \text{confidence} \times \text{probability} \]

For the purpose of weighing static analysis results, an accuracy measure is associated with each rule applied by the analysis engine. This number represents the possibility that the rule will correctly identify a vulnerability.

For example, the rule that Static Code Analysis uses to identify the member field race condition has a high accuracy because it precisely identifies assignments to a member field of a singleton object. Conversely, the rule used to identify cross-site request forgery has a low accuracy because it identifies potentially vulnerable form submissions and relies on a human auditor to determine whether the form submissions are susceptible to cross-site request forgery.

During static analysis, the analysis engine might have to make assumptions about the way the code will behave at runtime. The more assumptions the analysis engine makes, the more likely it is that a result will be incorrect.

The term confidence is used to estimate the possibility that the analysis engine correctly applies the rule. For example, Static Code Analysis reports reflected cross-site scripting vulnerabilities in a JSP where data from a request parameter is echoed directly to the page with high confidence. Conversely, Static Code Analysis reports persistent cross-site scripting issue where data read from a database into a class selected at runtime using dependency injection is rendered in the presentation tier with low confidence.

To represent the probability that the vulnerability will be discovered and acted upon (with action potentially coming the form of an exploit), Static Code Analysis associates a probability measure with each category of vulnerability identified by the rules. For example, cross-site scripting vulnerabilities carry a higher probability than member field race conditions because they are more likely to be discovered and exploited.

From a programmer’s perspective, some bugs are harder to fix than others. Modifying a single line of code in a self-contained method is easier than modifying the result of a sequence of calls that span the program. We use the term remediation effort to describe the relative amount of effort required to fix and verify a finding.

Static Code Analysis provides a remediation effort with each finding it reports. For example, member field race conditions have a small remediation effort, while cross-site request forgery, which often involves major changes to a website, has a high remediation effort.

To avoid implying too much precision where little exists, Static Code Analysis limits values of impact, accuracy, confidence, and probability to a decimal range of from 0.1 to 5.0 and scale the calculated likelihood value to the same range. It then defines high values for impact and likelihood as those at 2.5 and above [2.5,5.0] and low values as those below 2.5 (0,2.5).

Static Code Analysis does not provide units for remediation effort because the absolute cost of remediating different vulnerabilities will differ from one organization to another. Instead remediation effort estimates the
Appendix B: Static Analysis Results Prioritization

100

relative cost to remediate one kind of finding versus another, thereby enabling a comparison of the effort required to remediate different vulnerabilities or vulnerabilities across more than one project.

The following table provides sample impact, accuracy, confidence, and probability values for the four vulnerabilities mentioned in this section along with the resulting risk calculations and corresponding remediation effort for vulnerability category.

<table>
<thead>
<tr>
<th>Category</th>
<th>Impact</th>
<th>Accuracy</th>
<th>Confidence</th>
<th>Probability</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Race Condition: Singleton Member Field</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>Impact = 4 (High)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Likelihood = (5 · 5 · 3)/25 = 3 (High)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Risk = Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Estimated remediation effort = 5</td>
</tr>
<tr>
<td>Cross-Site Request Forgery</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>Impact = 2 (Low)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Likelihood = (1 · 5 · 2)/25 = &lt;1 (Low)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Risk = Low</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Estimated remediation effort = 12</td>
</tr>
<tr>
<td>Cross-Site Scripting: Reflected</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>Impact = 5 (High)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Likelihood = (5 · 5 · 5)/25 = 5 (High)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Risk = Critical</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Estimated remediation effort = 1</td>
</tr>
<tr>
<td>Cross-Site Scripting: Persistent</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>5</td>
<td>Impact = 5 (High)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Likelihood = (5 · 1 · 5)/25 = 1 (Low)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Risk = Medium</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Estimated remediation effort = 1</td>
</tr>
</tbody>
</table>