Micro Focus
Fortify WebInspect Tools

Software Version: 19.2.0
Windows® operating systems

Tools Guide for Fortify WebInspect Products

Document Release Date: November 2019

Software Release Date: November 2019

Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

https://www.microfocus.com

Warranty

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus 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. Except as specifically indicated otherwise, a valid license from Micro Focus is 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

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

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

This document was produced on October 22, 2019. To check for recent updates or to verify that you are using the most recent edition of a document, go to:

https://www.microfocus.com/support-and-services/documentation

About this PDF Version of Online Help

This document is a PDF version of the online help. This PDF file is provided so you can easily print multiple topics from the help information or read the online help in PDF format. Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online help.

Contents

Preface

Contacting Micro Focus Fortify Customer Support

If you have questions or comments about using this product, contact Micro Focus Fortify Customer Support using one of the following options.

To Manage Your Support Cases, Acquire Licenses, and Manage Your Account

https://softwaresupport.softwaregrp.com

To Call Support

1.844.260.7219

For More Information

For more information about Fortify software products:
https://software.microfocus.com/solutions/application-security

About the Documentation Set

The Fortify Software documentation set contains installation, user, and deployment guides for all Fortify Software products and components. In addition, you will find technical notes and release notes that describe new features, known issues, and last-minute updates. You can access the latest versions of these documents from the following Micro Focus Product Documentation website:

https://www.microfocus.com/support-and-services/documentation

Change Log

The following table lists changes made to this document. Revisions to this document are published between software releases only if the changes made affect product functionality.

Software Release / Document Version	Changes
19.2.0	Added: • Chapter covering the new Web Macro Recorder with Macro Engine 5.0. For more information, see "Web Macro Recorder with Macro Engine 5.0" on page 221. Updated: • Previous version of Web Macro Recorder to include supported macro engine version. See "Web Macro Recorder with Macro Engine 4.0" on page 172.
19.1.0	Updated: • Parameters Editor topic to include information about multi-login scans. See "Parameters Editor" on page 200. • List of policies with description of the WebSocket policy. See "Policies" on page 79. • Various topics to show that Custom Agents and Custom Checks have been moved from the Attack Groups category to the Seven Pernicious Kingdoms category. See the following topics: • "About the Policy Manager Tool" on page 56 • "Creating a Custom Check" on page 60 • "Using a Custom Agent" on page 70 Removed: • Support Tool chapter.
18.20	Updated: • List of policies with descriptions for General Data Protection Regulation (GDPR), SANS Top 25, and DISA STIG. See "Policies" on page 79.

Chapter 1: Welcome to Micro Focus Fortify WebInspect Tools

About Fortify WebInspect Tools

Fortify WebInspect Tools is a robust set of diagnostic and penetration testing tools and configuration utilities packaged with Fortify WebInspect and Micro Focus Fortify WebInspect Enterprise.

The tools provided in Fortify WebInspect Enterprise are a subset of the tools provided in Fortify WebInspect. The chapters in this guide that describe tools that are provided in Fortify WebInspect but not in Fortify WebInspect Enterprise have titles that end with "(Fortify WebInspect Only)."

Using Tools with a Proxy

When using tools that incorporate a proxy, you may encounter servers that do not ask for a client certificate even though a client certificate is required. To accommodate this situation, you must edit the `SPI.Net.Proxy.Config` file.

Related Documents

This topic describes documents that provide information about Micro Focus Fortify software products.

Note: You can find the Micro Focus Fortify Product Documentation at https://www.microfocus.com/support-and-services/documentation. All guides are available in both PDF and HTML formats. Product help is available within the Fortify WebInspect products.

All Products

The following documents provide general information for all products. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website.

Document / File Name	Description
About Micro Focus Fortify Product Software Documentation About_Fortify_Docs_<*version*>.pdf	This paper provides information about how to access Micro Focus Fortify product documentation. **Note:** This document is included only with the product

Document / File Name	**Description**
	download.
Micro Focus Fortify Software System Requirements Fortify_Sys_Reqs_<*version*>.pdf	This document provides the details about the environments and products supported for this version of Fortify Software.
Micro Focus Fortify Software Release Notes FortifySW_RN_<*version*>.pdf	This document provides an overview of the changes made to Fortify Software for this release and important information not included elsewhere in the product documentation.
What's New in Micro Focus Fortify Software <version> Fortify_Whats_New_<*version*>.pdf	This document describes the new features in Fortify Software products.

Micro Focus Fortify WebInspect

The following documents provide information about Fortify WebInspect. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-webinspect.

Document / File Name	**Description**
Micro Focus Fortify WebInspect Installation Guide WI_Install_<*version*>.pdf	This document provides an overview of Fortify WebInspect and instructions for installing Fortify WebInspect and activating the product license.
Micro Focus Fortify WebInspect User Guide WI_Guide_<*version*>.pdf	This document describes how to configure and use Fortify WebInspect to scan and analyze Web applications and Web services. **Note:** This document is a PDF version of the Fortify WebInspect help. This PDF file is provided so you can easily print multiple topics from the help information or read the help in PDF format. Because this content was originally created to be viewed as help in a web browser, some topics may not be formatted properly. Additionally, some interactive topics and linked content may not be present in this PDF version.

Document / File Name	Description
Micro Focus Fortify WebInspect on Docker User Guide WI_Docker_Guide_<*version*>.pdf	This document describes how to download, configure, and use Fortify WebInspect that is available on the Docker container platform. This full version of the product is intended to be used in automated processes as a headless scanner configured by way of the command line interface (CLI) or the application programming interface (API).
Micro Focus Fortify WebInspect Tools Guide WI_Tools_Guide_<*version*>.pdf	This document describes how to use the Fortify WebInspect diagnostic and penetration testing tools and configuration utilities packaged with Fortify WebInspect and Fortify WebInspect Enterprise.
Micro Focus Fortify WebInspect Runtime Agent Installation Guide WI_RT_Agent_Install_<*version*>.pdf	This document describes how to install the Fortify WebInspect Runtime Agent for applications running under a supported Java Runtime Environment (JRE) on a supported application server or service and applications running under a supported .NET Framework on a supported version of IIS.
Micro Focus Fortify WebInspect Agent Rulepack Kit Guide WI_Agent_Rulepack_Guide_<*version*>.pdf	This document describes the detection capabilities of Fortify WebInspect Agent Rulepack Kit. Fortify WebInspect Agent Rulepack Kit runs atop the Fortify WebInspect Runtime Agent, allowing it to monitor your code for software security vulnerabilities as it runs. Fortify WebInspect Agent Rulepack Kit provides the runtime technology to help connect your dynamic results to your static ones.

Micro Focus Fortify WebInspect Enterprise

The following documents provide information about Fortify WebInspect Enterprise. Unless otherwise noted, these documents are available on the Micro Focus Product Documentation website at https://www.microfocus.com/documentation/fortify-webinspect-enterprise.

Document / File Name	Description
Micro Focus Fortify WebInspect Enterprise Installation and Implementation Guide	This document provides an overview of Fortify WebInspect Enterprise and instructions for installing Fortify WebInspect Enterprise, integrating it with Fortify Software Security

Document / File Name	**Description**
WIE_Install_<*version*>.pdf	Center and Fortify WebInspect, and troubleshooting the installation. It also describes how to configure the components of the Fortify WebInspect Enterprise system, which include the Fortify WebInspect Enterprise application, database, sensors, and users.
Micro Focus Fortify WebInspect Enterprise User Guide WIE_Guide_<*version*>.pdf	This document describes how to use Fortify WebInspect Enterprise to manage a distributed network of Fortify WebInspect sensors to scan and analyze Web applications and Web services. **Note:** This document is a PDF version of the Fortify WebInspect Enterprise help. This PDF file is provided so you can easily print multiple topics from the help information or read the help in PDF format. Because this content was originally created to be viewed as help in a web browser, some topics may not be formatted properly. Additionally, some interactive topics and linked content may not be present in this PDF version.
Micro Focus Fortify WebInspect Tools Guide WI_Tools_Guide_<*version*>.pdf	This document describes how to use the Fortify WebInspect diagnostic and penetration testing tools and configuration utilities packaged with Fortify WebInspect and Fortify WebInspect Enterprise.

Chapter 2: About the Audit Inputs Editor Tool

This tool allows you to create or edit inputs to the audit engines and to a distinct set of checks.

There are two ways to access the Audit Inputs Editor:

- From the Policy Manager (using the Policy Manager **Tools** menu). Use this method to create or modify an inputs file (<*filename*>`.inputs`). You can then specify this file when modifying scan settings.

  To modify an inputs file, click the **Open** icon on the Audit Input Editor's toolbar or select **File > Open**.
- From the Default or Current Settings, by clicking the **Audit Inputs Editor** button on the Attack Exclusions settings. Using this method, you can modify the Default settings file directly, but you cannot create a separate inputs file.

If you access the Audit Inputs Editor from Default Settings or Current Settings, the check inputs you create or modify become part of the settings file.

However, if you access the Audit Inputs Editor from the Policy Manager, you must import into Fortify WebInspect the saved file containing your check input modifications, as follows:

1. On the Fortify WebInspect menu bar, click **Edit > Default Settings**.
2. Under **Audit Settings**, select **Attack Exclusions**.
3. Click **Import Audit Inputs**.
4. Select the file you created (`*.inputs`) and click **Open**.

When accessed through the Current Settings window or the Default Settings window, Attack Exclusions panel, the Audit Inputs Editor does not contain a menu bar or toolbar.

Check Inputs

Certain checks require inputs that accommodate the specific design of the target website. Fortify WebInspect conducts these checks using default values, which you may need to change.

To create or modify inputs for specific checks:

1. Click the **Check Inputs** tab.
2. Select a check from the list.

   The inputs for the selected check appear on the right.
3. Enter the requested input values.
4. Do one of the following:
   - If you launched the Audit Inputs Editor from the Default Settings or Current Settings, click **OK**.

- If you launched the Audit Inputs Editor from the Policy Manager, click **File > Save** or **File > Save As**.

See Also

"Engine Inputs" below

Engine Inputs

To create or modify inputs to audit engines:

1. Click the **Engine Inputs** tab.
2. Click the drop-down arrow.
   a. To apply your modifications to all audit engines, select **<Default>**. The Default parameters are extracted from the default Fortify WebInspect Audit Settings - Attack Exclusions.
   b. To modify inputs for a specific audit engine, select one from the list.
3. Select an engine input.
4. If you selected one of the following:
   - Excluded Query Parameters
   - Excluded Post Parameters
   - Excluded Cookies
   - Excluded Headers
   - Root Directories

   then do the following:
   - To add an item to the list, click **Add**.
   - To edit an item, select an item and click **Edit**.
   - To delete an item, select the item and click **Remove**.
   - If you selected a specific engine (rather than Defaults), select one of the following options:
     - **Merge with defaults** - The parameters you specified are added to the Defaults list, which apply to all engines.
     - **Replace defaults** - The engine will use the parameters you specified instead of those in the Defaults list.

   Note: If you specify a Root Directory, then the engine will attack the object in the directory you specify, rather than the actual root. For example, if an engine normally attacks filename.txt in the default root directory *rootdir* (/rootdir/filename.txt), then if you specify a root directory of */foobar/*, the engine will attack /foobar/filename.txt.
5. If you selected one of the following:

- Header Audit Rules
- Cookie Audit Rules

then do the following:

a. Clear the **Use value from defaults** check box.

b. Select an option from the drop-down list. Options are as follows:

   Header Audit Rules

   - **Attack All Every Time** - Attack the header in every request.
   - **Attack Once Per Directory** - Attack each named header in every directory only once the first time it is encountered.
   - **Attack Only Once** - Attack the header only once per host the first time it is encountered during the scan.

   Cookie Audit Rules

   - **Attack All** - Attack all cookies that are encountered in every request during the scan.
   - **Attack Only Cookies In Children Set In Parent** - Attack the inherited cookie in every child session in which it is encountered.

     For example, if the parent session request sets the following cookie with JSESSION ID:

     ```
     GET /auth/link.page; HTTP/1.1
     Referer: http://zero.webappsecurity.com/auth/security-check.html
     …
     Cookie:
     CustomCookie=WebInspect83644ZX632F0EE21C7249358BE159C67CEE9085YCE
     51;
     JSESSIONID=2DC913EA;username=username;password=password
     ```

     And the child session includes the inherited cookie:

     ```
     GET /auth/link.page HTTP/1.1
     Referer: http://zero.webappsecurity.com/auth/link.page;
     …
     Cookie:
     CustomCookie=WebInspect83644ZX632F0EE21C7249358BE159C67CEE9085YCE
     51;
     JSESSIONID=2DC913EA;username=username;password=password
     ```

     Then the cookie will be attacked in the child session.

     A child session might have multiple cookies, but only the one that was set in the parent session will be attacked.

   - **Attack Each Cookie Once** - Attack each unique cookie only once per host the first time it is encountered during the scan.

6. Click **OK** if you launched the Audit Inputs Editor from Default or Current Settings, or click **File > Save** or **File > Save As** if you launched the Audit Inputs Editor from the Policy Manager.

See Also

"Check Inputs" on page 25

Chapter 3: About the Compliance Manager Tool (Fortify WebInspect Only)

Fortify WebInspect employs an extensive arsenal of attack agents designed to detect security flaws in web-based applications. It probes your system with thousands of HTTP requests and evaluate each individual response. This session-based assessment reports each vulnerability, pinpoints its location in the application, and recommends corrective actions you should take. It is, basically, a quantitative analysis of your system.

Fortify WebInspect can also perform a qualitative analysis by grading how well your application complies with certain government-mandated regulations or corporate-defined guidelines. For example, the Health Insurance Portability and Accountability Act (HIPAA) requires healthcare providers using web-based applications to provide "procedures for creating, changing, and safeguarding passwords." With Fortify WebInspect, you can assess your application and then generate a Compliance Report that measures how well your application satisfies this HIPAA rule.

How It Works

You create a compliance template that associates requirements with one or more attack agents or vulnerabilities. For example, you might include the statement (or question) "The application will not use any 'hidden' fields." The attack agent that tests for compliance to this requirement is Hidden Form Value, ID #4727 (which is one of the agents in the "General Text Searching Group" on page 35).

Compliance templates are completely flexible. You can enable or disable individual requirements. You can also modify requirements by adding or removing attack agents or "Threat Classes" on page 35. For maximum flexibility, you can even create your own agents and associate them with a user-defined requirement.

Fortify WebInspect includes sample compliance templates that you can edit to fit your company's specific requirements.

For step-by-step instructions for creating a policy, see "Creating a Compliance Template" on the next page.

To test your website for compliance:

1. If necessary, create or modify a compliance template.
2. Scan your website.
3. On the Fortify WebInspect **Start** page, click **Generate a Report**.

   The Generate a Report window opens.
4. If the scan data is stored in a different database, click **Change DB** and then select a database.
5. Select a scan (designated by name, URL, or IP address).
6. Click **Next**.

7. Select **Compliance**.
8. If you want to produce individual reports on separate tabs (rather than combining all reports on one tab), select **Open Reports in Separate Tabs**.
9. Select either **Adobe PDF** or **HTML** as the report format.

   Adobe Reader 7 or later is required to read reports in portable data format (PDF).
10. Specify a compliance template. You can select a default template from the list, click the browse button to browse for templates you have created, or open the Compliance Manager and create a custom template.
11. Click **Finished**.
12. After Fortify WebInspect generates the report and displays it on a tab, you can save a report by clicking the Save Report icon on the toolbar.

See Also

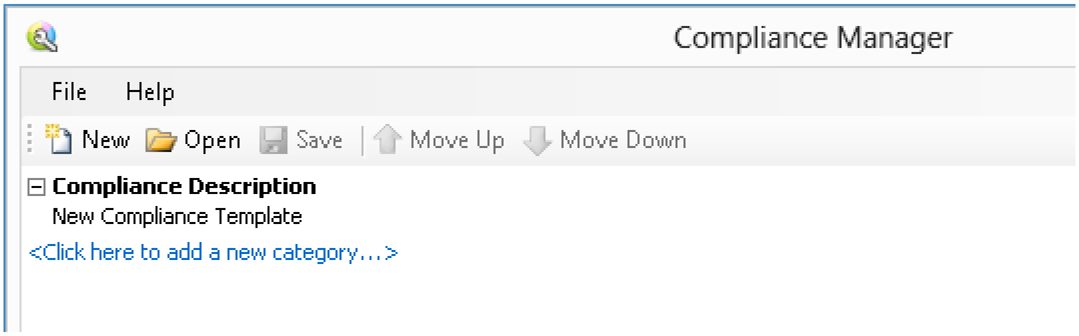
"Creating a Compliance Template" below

Creating a Compliance Template

To create a compliance template:

1. On the Fortify WebInspect menu bar, click **Tools > Compliance Manager**.

   The Compliance Manager window opens, displaying the outline of a new template.

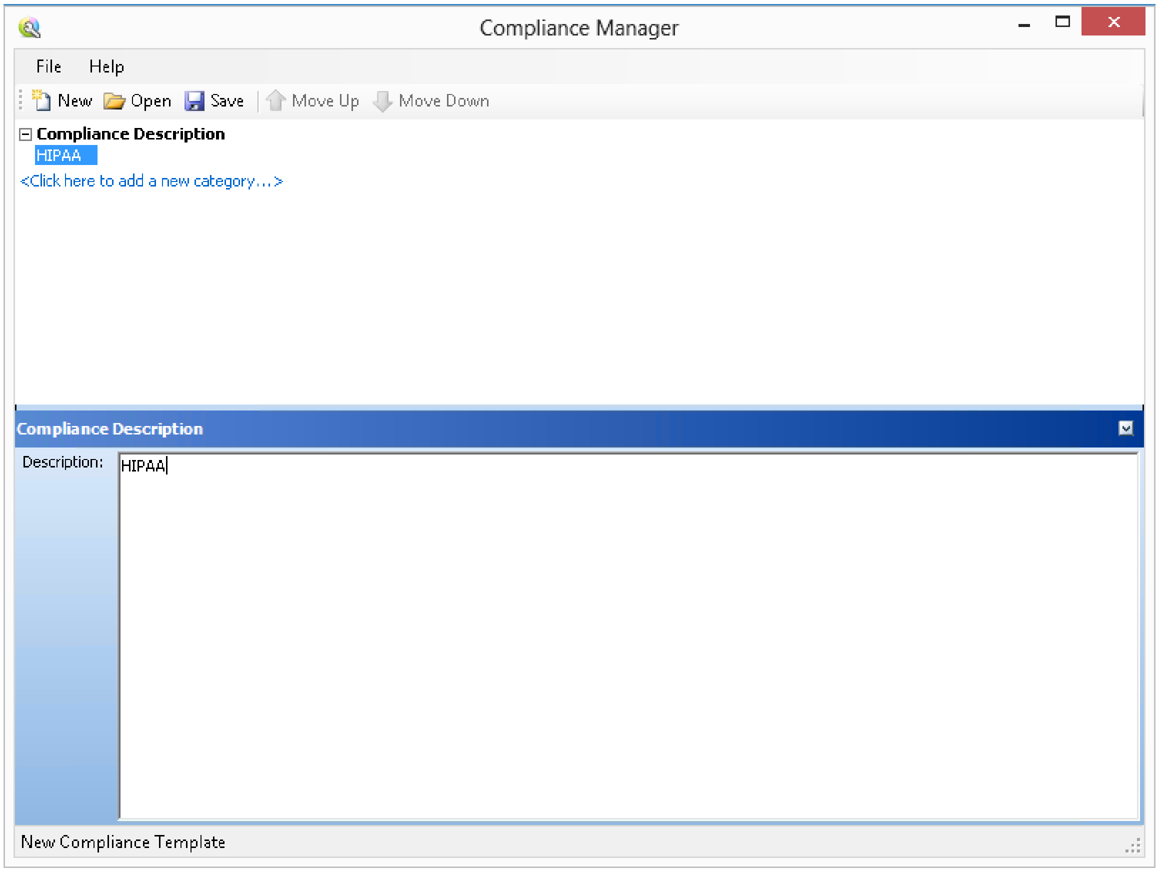


Compliance Description
New Compliance Template
<Click here to add a new category...>

2. Click the phrase "New Compliance Template."

   The Compliance Manager creates an editing area in the lower half of the window.

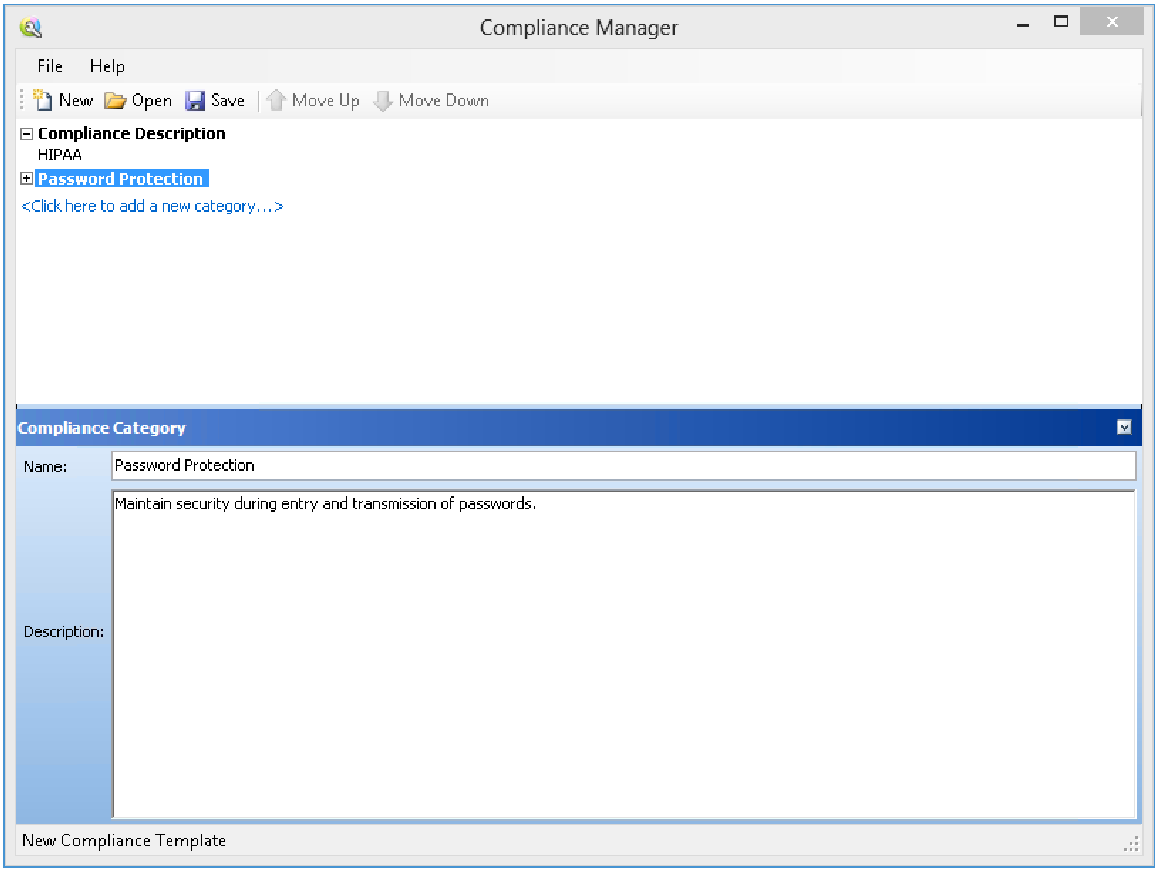
3. In the editing area, replace the phrase "New Compliance Template" with a description of the template you are creating ("HIPAA" in this example).



Description: HIPAA

4. Click the phrase "<Click here to add a new category...>."

5. In the editing area, enter the name and description of the new category. In this example, the name is "Password Protection" and the description is "Maintain security during entry and transmission of passwords."



Name: Password Protection

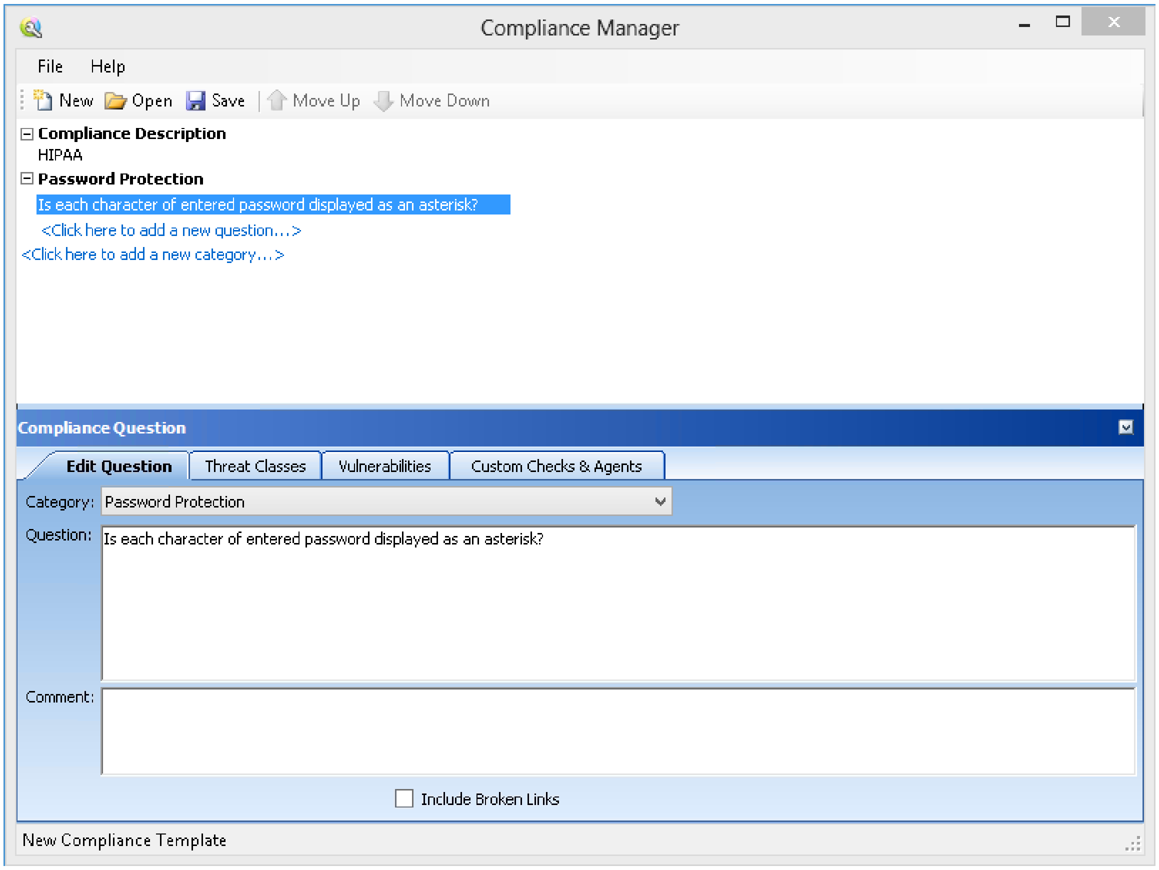
Description: Maintain security during entry and transmission of passwords.

New Compliance Template

6. Click the plus sign ⊞ to expand the node labeled Password Protection.
7. Click the phrase "<Click here to add a new question...>."
8. Click the phrase "New Question."

   The editing area displays tabs allowing you to create a question related to the category "Password Protection."

9. In the **Question** area, type a question related to the category. This example asks the question, "Is each character of entered password displayed as an asterisk?"



Compliance Description
HIPAA
Password Protection
Is each character of entered password displayed as an asterisk?
<Click here to add a new question...>
<Click here to add a new category...>

Category: Password Protection
Question: Is each character of entered password displayed as an asterisk?
Comment:
□ Include Broken Links

New Compliance Template

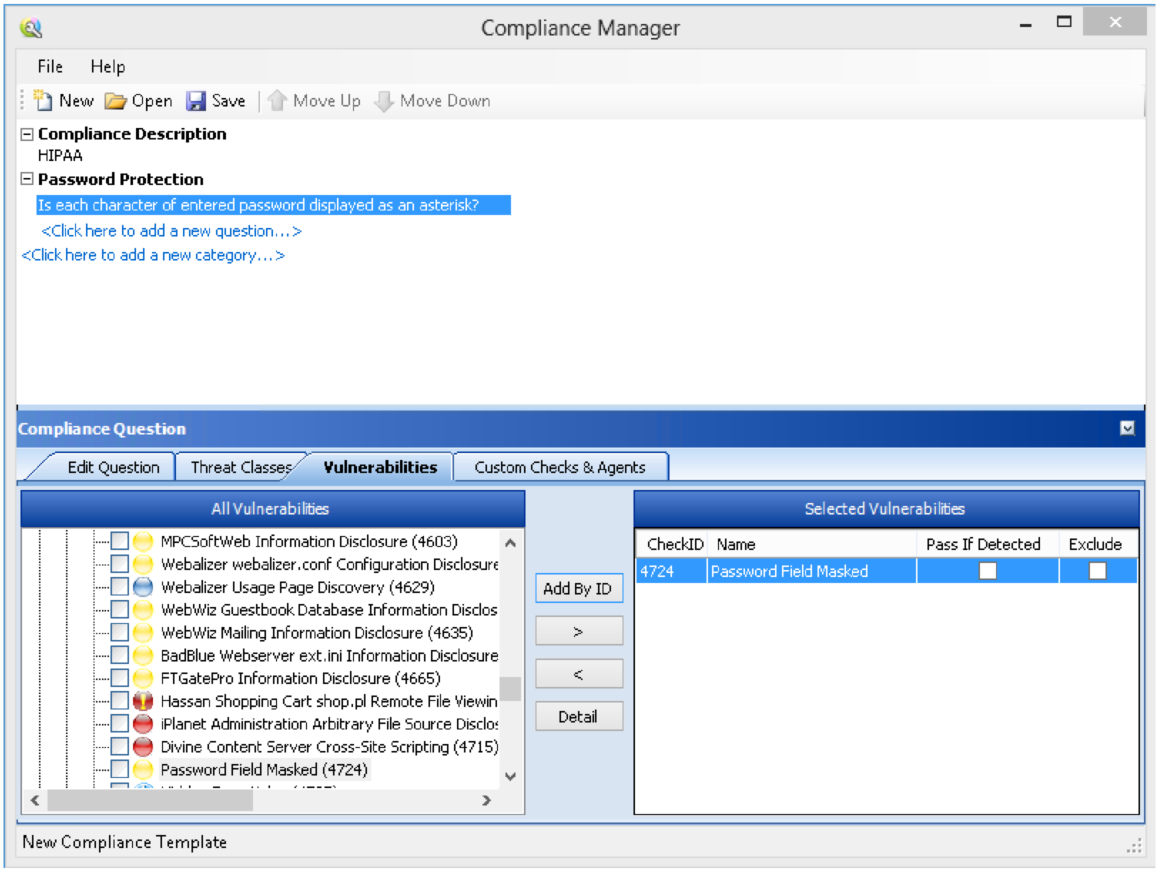
10. You can associate this question with threat classes, vulnerabilities defined by Micro Focus, or a custom check or agent that you previously created. For this example, click the **Vulnerabilities** tab and then click **Add By ID**.

Note: You can also select a vulnerability or a threat class and click ▶ to include it in the **Selected Vulnerabilities** or **Selected Threat Classes** section for this question.

11. On the Add Check By ID window, enter 4724 and click **OK**. 4724 is the ID number of the "Password Field Masked" check.

Note: You can add multiple IDs (one per line).

The check you specified appears in the **Selected Vulnerabilities** area.



⊟ Compliance Description
HIPAA
⊟ Password Protection
Is each character of entered password displayed as an asterisk?
<Click here to add a new question...>
<Click here to add a new category...>

- □ MPCSoftWeb Information Disclosure (4603)
- □ Webalizer webalizer.conf Configuration Disclosure
- □ Webalizer Usage Page Discovery (4629)
- □ WebWiz Guestbook Database Information Disclos
- □ WebWiz Mailing Information Disclosure (4635)
- □ BadBlue Webserver ext.ini Information Disclosure
- □ FTGatePro Information Disclosure (4665)
- □ Hassan Shopping Cart shop.pl Remote File Viewin
- □ iPlanet Administration Arbitrary File Source Disclo:
- □ Divine Content Server Cross-Site Scripting (4715)
- □ Password Field Masked (4724)

CheckID	Name	Pass If Detected	Exclude
4724	Password Field Masked	□	□

New Compliance Template

12. The **Selected Vulnerabilities** area contains two check boxes:

    - **Pass If Detected** - Select this option if the check is designed to confirm an attribute that contributes to application security. You might use this if, for example, you develop a custom check that checks for the existence of a file (such as Privacy Policy.html) that is part of your compliance program.
    - **Exclude** - Select this option if you add a group of checks, but want to exclude specific ones.

    In this example, do not select either check box.

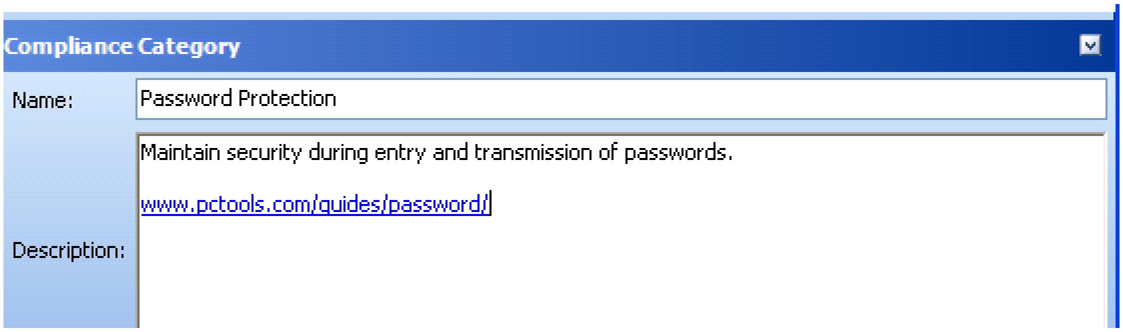
13. To view a list of broken links in the compliance report, select the **Include Broken Links** check box.

    If you select the check box, then when you run a compliance report, any broken links found will be listed at the end of the report. If broken links are associated with a question in the template, then that question will be marked as failed.

14. Continue adding threat classes, vulnerabilities, or custom checks until you have included all that sufficiently test your application for the compliance question.
15. Create additional questions and categories using the above procedures until the compliance template is complete.
16. Click **Save**.

Usage Notes

- To rearrange categories or items, select an item and click **Move Up** or **Move Down**.
- To insert categories or items, you can alternatively right-click a category/question and select **Insert** from the shortcut menu. The item will be inserted above the selected item.
- You can add an HTML link to any description or question, as depicted in the following illustration.



Name: Password Protection

Description: Maintain security during entry and transmission of passwords.

www.pctools.com/guides/password/

General Text Searching Group

This group of agents, used mainly by the Directory Enumeration engine, follows all known and unknown paths located on your site. Individual checks are grouped alphabetically from A (which begins with the search for a directory named Accounting) to Z (which ends with the search for a directory named Zips). This group also includes checks for other types of commonly occurring directories, such as those associated with Microsoft FrontPage and Microsoft Internet Information Server log files (W3SVCnn).

For detailed information about all the possible agents, start the Policy Manager in Standard view, expand the General Text Searching node and click on any agent.

Threat Classes

The Web Application Security Consortium has developed industry-standard terminology to clarify and organize threats to the security of a web site. These are listed on the **Threat Classes** tab.

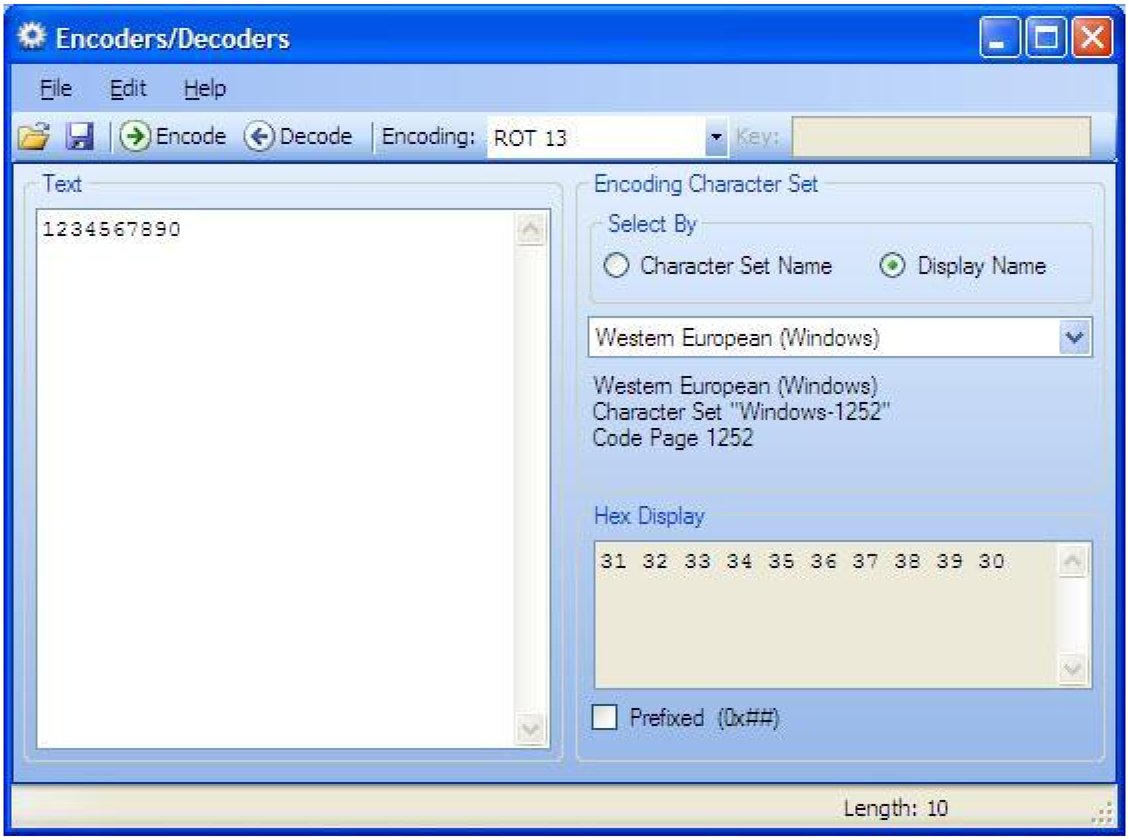
To determine if a scan revealed a susceptibility to these threats:

1. Select a threat class (or one of its components).
2. Click ▶ to include it in the **Selected Threat Classes** for this question.

Chapter 4: About the Encoders/Decoders Tool

This tool allows you to encode and decode values using Base 64, hexadecimal, MD5, and other schemes. You can also encode a string into a Unicode string and use special characters in URL construction.

During the analysis of your scan results, when you encounter a string that you suspect is in an encoded or encrypted format, you can simply copy the string, paste it into the Encoders/Decoders tool, and then click **Decode**.



Encoding: ROT 13

Key:

1234567890

○ Character Set Name ● Display Name

Western European (Windows)

Western European (Windows)
Character Set "Windows-1252"
Code Page 1252

```
31 32 33 34 35 36 37 38 39 30
```

□ Prefixed (0x##)

Length: 10

Encoding a String

To encode a string:

1. Type (or paste) a string in the **Text** area, or load the contents of a file by selecting **File > Open** from the menu.
2. Select an encoding character set using either the **Character Set Name** or the **Display Name**.
3. Select a cipher type from the **Encoding** list. For more information, see "Encoding Types" on

page 38.

4. If necessary, type a key in the **Key** field. When a valid key is entered, the **Encode** and **Decode** buttons become enabled.
5. Click **Encode**.

   The **Text** area displays the encoded string. The **Hex Display** area displays the hexadecimal value of each character in the encoded string (formatted in the character set that you select).

   If you select **Prefixed**, "0x" is added to the beginning of the hexadecimal numbers. C and languages with a similar syntax (such as C++, C#, Java and JavaScript) prefix hexadecimal numerals with “0x” (for example, 0x5A3). The leading zero allows the parser to recognize a number, and the “x” stands for hexadecimal.

Decoding a String

To decode a string:

1. Type (or paste) a string in the **Text** area, or load the contents of a file by selecting **File > Open** from the menu.
2. Select a cipher type from the **Encoding** list.
3. If necessary, type a key in the **Key** field.
4. Click **Decode**.

You can also use Fortify WebInspect's encoding and decoding capabilities in the HTTP Editor. Right-click while editing a session to access encoding and decoding options.

Manipulating Encoded Strings

The encoded form of a string may contain characters that are non-printable. This often occurs when using a hash-based encoding scheme or any encoding scheme that requires a key. Since non-printable characters cannot be copied to the Windows clipboard, you cannot simply copy from or paste into the Encoder/Decoder. However, there are two methods you can use to work around this limitation:

- Save the encoded string to a file and, when you want to decode it, select **File > Open** from the menu to load it into the Encoder tool. Then decode it using the original method and (if applicable) key.
- Also, after encoding the string using the chosen encoding method and key, you can encode the resulting string using the base 64 method; then copy the string to the clipboard, paste the clipboard contents, decode using base 64, and decode again using the original method and (if applicable) key.

Encoding Types

The Encoder/Decoder allows you to select the encoding types described in the following table.

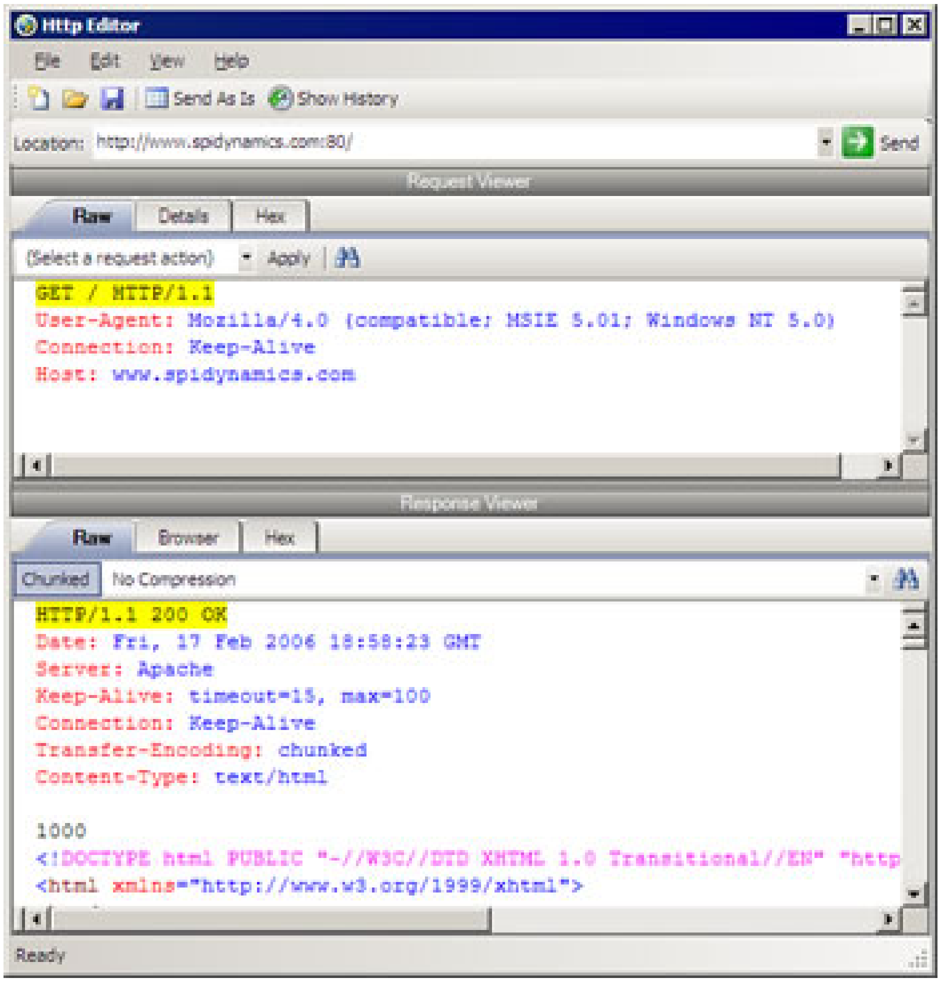
Encoding Type	Definition
3DES	Triple DES; a mode of the DES encryption algorithm that encrypts data three times (the string is encrypted, then the encryption is encrypted, and the resulting cipher text is encrypted a third time). The key must be 128 or 192 bits (16 or 24 characters).
Base64	Encodes and decodes triplets of 8-bit octets as groups of four characters, each representing 6 bits of the source 24 bits. Only characters present in all variants of ASCII and EBCDIC are used, avoiding incompatibilities in other forms of encoding.
Blowfish	An encryption algorithm that can be used as a replacement for the DES algorithm.
DES	Data Encryption Standard. A widely used method of data encryption that can use more than 72 quadrillion different private (and secret) encryption keys. Both the sender and the user must use the same private key.
Hex	Hexadecimal.
MD5	Produces a 128-bit "fingerprint" or "message digest" of whatever data you enter.
RC2	A variable key-size block cipher designed by Ronald Rivest. It has a block size of 64 bits and is about two to three times faster than DES in software.
RC4	A stream cipher designed by Ronald Rivest. It is a variable key-size stream cipher with byte-oriented operations. Used for file encryption in products such as RSA SecurPC and also used for secure communications, as in the encryption of traffic to and from secure web sites using the SSL protocol.
ROT13	A simple Caesar cipher used for obscuring text by replacing each letter with the letter thirteen places down the alphabet.
SHA1	Secure Hash Algorithm. A one-way hash function developed by NIST and defined in standard FIPS 180. SHA-1 is a revision published in 1994; it is also described in ANSI standard X9.30 (part 2).
SHA256	Secure Hash Algorithm that uses 256-bit encryption.
SHA384	Secure Hash Algorithm that uses 384-bit encryption.

Encoding Type	Definition
SHA512	Secure Hash Algorithm that uses 512-bit encryption.
ToLower	Changes uppercase letters to lowercase.
ToUpper	Changes lowercase letters to uppercase.
TwoFish	An encryption algorithm based on an earlier Blowfish.
Unicode	Provides a unique number for every character, regardless of the platform, program, or language.
URL	Creates values that can be used for URL-encoding non-standard letters and characters for display in browsers and plug-ins that support them.
XHTML	Encapsulates the entered data with text tags: <text>data</text>
XOR	XOR performs an Exclusive OR operation. You must provide a key. If the length of the key string is only one character, that character is ORed against each character in the encode/decode string.

Chapter 5: About the HTTP Editor Tool

Use the HTTP Editor to create or edit requests, send them to a server, and view the response either in raw HTML or as rendered in a browser. The HTTP Editor is a manual hacking tool, and requires a working knowledge of HTML, HTTP, and request methods.

To set proxy and authorization parameters, if necessary, select **Edit > Settings**.



http://www.spidynamics.com:80/

```
GET / HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Connection: Keep-Alive
Host: www.spidynamics.com
```

```
HTTP/1.1 200 OK
Date: Fri, 17 Feb 2006 18:58:23 GMT
Server: Apache
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html

1000
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http
<html xmlns="http://www.w3.org/1999/xhtml">
```

Request Viewer

The Request Viewer contains the HTTP request message, which you can view in four different formats using the following tabs:

- **Raw** - Depicts the line-by-line textual format of the request message.
- **Details** - Displays the header names and field values in a table format.
- **Hex** - Displays the hexadecimal and ASCII representation of the message.

- **XML** - Displays any XML content in the message body. (This tab appears only if the request contains XML-formatted data.)

Response Viewer

The Response Viewer contains the HTTP response message, which you can also view in four different formats using the following tabs:

- **Raw** - Depicts the line-by-line textual format of the response message.
- **Browser** - Displays the response message as rendered in a browser.
- **Hex** - Displays the hexadecimal and ASCII representation of the response message.
- **XML** - Displays any XML content in the message body. (This tab appears only if the response contains XML-formatted data.)

HTTP Editor Menus

File Menu

The **File** menu contains the following options:

- **New Request** - Deletes all information from previous sessions and resets the Location URL.
- **Open Request** - Allows you to load a file containing an HTTP request saved during a previous session.
- **Save Request** - Allows you to save an HTTP request.
- **Save Request As** - Allows you to save an HTTP request.
- **URL Synchronization** - When selected, any characters you type into the **Address** combo box are added to the Request-URI of the HTTP request line.
- **Send As Is** - If you select this option, the HTTP Editor will not modify the request, regardless of any other settings you may select. This allows you to send a purposely malformed message. Authentication and proxy settings are disabled when using this option.

  > **Note:** You may manually edit the request to go through a proxy, but many standard HTTP proxy servers cannot process non-compliant HTTP requests.

- **Exit** - Closes the HTTP Editor.

Edit Menu

The **Edit** menu contains the following options:

- **Cut** - Deletes selected text and saves it to the clipboard.
- **Copy** - Saves the selected text to the clipboard.
- **Paste** - Inserts text from the clipboard
- **Find** - Displays a dialog box that allows you to search for text that you specify.

- **Settings** - Allows you to configure request, authentication, and proxy parameters for the HTTP Editor.

View Menu

The **View** menu contains the following options:

- **Show History** - Displays a pane listing all HTTP requests sent.
- **Word Wrap** - Causes all text to fit within the defined margins.

Help Menu

The **Help** menu contains the following commands:

HTTP Editor Help - Opens the Help file with the **Contents** tab active.

Index - Opens the Help file with the **Index** tab active.

Search - Opens the Help file with the **Search** tab active.

About HTTP Editor - Displays information about the HTTP Editor.

Request Actions

The following options are available from the **Request Action** list in the Request Viewer pane.

PUT File Upload

The PUT method requests that the enclosed entity be stored under the supplied Request-URI.

To write a file to a server:

1. Select **PUT File Upload** from the drop-down list on the Request Viewer pane.
2. In the text box that appears to the right of the list, type the full path to a file

   - or -

   Click the Open Folder icon and select the file you want to upload.
3. Click **Apply**. This will also recalculate the content length.

Change Content-Length

In normal mode, if you edit the message body of the request, the HTTP Editor recalculates the content length and substitutes the appropriate value in the Content-length header. However, when using the **Send As Is** option, the HTTP Editor does not modify the content length. You can force this recalculation before sending the request by selecting **Change Content-Length** and clicking **Apply**.

URL Encode/Decode Param Values

The specification for URLs (RFC 1738, Dec . '94) limits the use of characters in URLs to a subset of the US-ASCII character set. HTML, on the other hand, allows the entire range of the ISO-8859-1 (ISO-Latin) character set to be used in documents, and HTML4 expands the allowable range to include the complete Unicode character set as well. To circumvent this limitation, you can encode non-standard letters and characters for display in browsers and plug-ins that support them.

URL encoding of a character consists of a "%" symbol, followed by the two-digit hexadecimal representation of the ISO-Latin code point for the character. For example:

- The asterisk symbol (*) = 42 decimal in the ISO-Latin set
- 42 decimal = 2A hexadecimal
- URL code for asterisk = %2A

You can use URL encoding to bypass an intruder detection system (IDS) that inspects request messages for certain keywords using only the ISO-Latin character set. For example, the IDS may search for "login" (in ISO-Latin), but not "%4C%4F%47%49%4E" (the URL-encoded equivalent).

To substitute URL code for parameters throughout the entire message, select **URL Encode Param Values** and click **Apply**.

To translate URL-encoded parameters to ISO-Latin, select **URL Decode Param Values** and click **Apply**.

Unicode Encode/Decode Request

The Unicode Worldwide Character Standard includes letters, digits, diacritics, punctuation marks, and technical symbols for all the world's principal written languages, using a uniform encoding scheme. Incorporating Unicode into client-server applications and websites offers significant cost savings over the use of legacy character sets. Unicode enables a single software product or a single website to be targeted across multiple platforms, languages and countries without re-engineering. It allows data to be transported through many different systems without corruption.

To translate the entire request message into Unicode, select **Unicode Encode Request** and click **Apply**.

To translate the entire request message from Unicode into ISO-Latin, select **Unicode Decode Request** and click **Apply**.

Create MultiPart Post

The POST method is used to request that the origin server accept the entity enclosed in the request as a new subordinate of the resource identified by the Request-URI in the Request-Line. You can attempt to upload data by manipulating a POST request message.

To insert data from a file:

1. Select **Create MultiPart Post** from the **Action** drop-down list on the Request Viewer pane.
2. In the text box to the right of the **Action** list, type the full path to a file

   - or -

   Click the Open Folder icon and select the file you want to insert.
3. Click **Apply**.

Remove MultiPart Post

To remove a file that is part of a multipart request, select **Remove MultiPart Post** from the **Action** list on the Request Viewer pane.

Response Actions

The area immediately below the tabs on the **Response Viewer** pane contains three controls:

- a **Chunked** button
- a **Content Coding** drop-down list
- a 🔍 button that launches the Find In Response dialog box, allowing you to search the response for the text string you specify.

Chunked

If a server starts sending a response before knowing its total length, it might break the complete response into smaller chunks and send them in series. Such a response contains the "Transfer-Encoding: chunked" header. A chunked message body contains a series of chunks, followed by a line with "0" (zero), followed by optional footers and a blank line. Each chunk consists of two parts:

- A line with the size of the chunk data, in hex, possibly followed by a semicolon and extra parameters you can ignore (none are currently standard), and ending with CRLF.
- The data itself, followed by CRLF.

Content Codings

If the HTTP response contains compressed data, you can decompress the data using one of the options from the list.

- GZIP - A compression utility written for the GNU project.
- Deflate - The "zlib" format defined in RFC 1950 [31] in combination with the "deflate" compression mechanism described in RFC 1951 [29].

See Also

"Editing and Sending a Request" below

"Searching the Request or Response" on page 46

Editing and Sending a Request

To edit and send a request:

1. Modify the request message in the Request Viewer pane.

   To encode or decode a text string, select the text, then right-click the selection and select either **Encoding** or **Decoding** from the pop-up menu.

   To change certain features of the request, select an item from the **Action** list and click **Apply**. See "About the HTTP Editor Tool" on page 40 for more information.
2. Click **Send** to send the HTTP request message.

   The Response Viewer pane displays the HTTP response message when it is received.
3. To view the response as rendered in a browser, click the **Browser** tab.

4. You can prepare your next HTTP request using the HTML or JavaScript controls rendered on the **Browser** tab. To use this feature, you must select the Interactive Navigation option (click **Edit > Settings**).

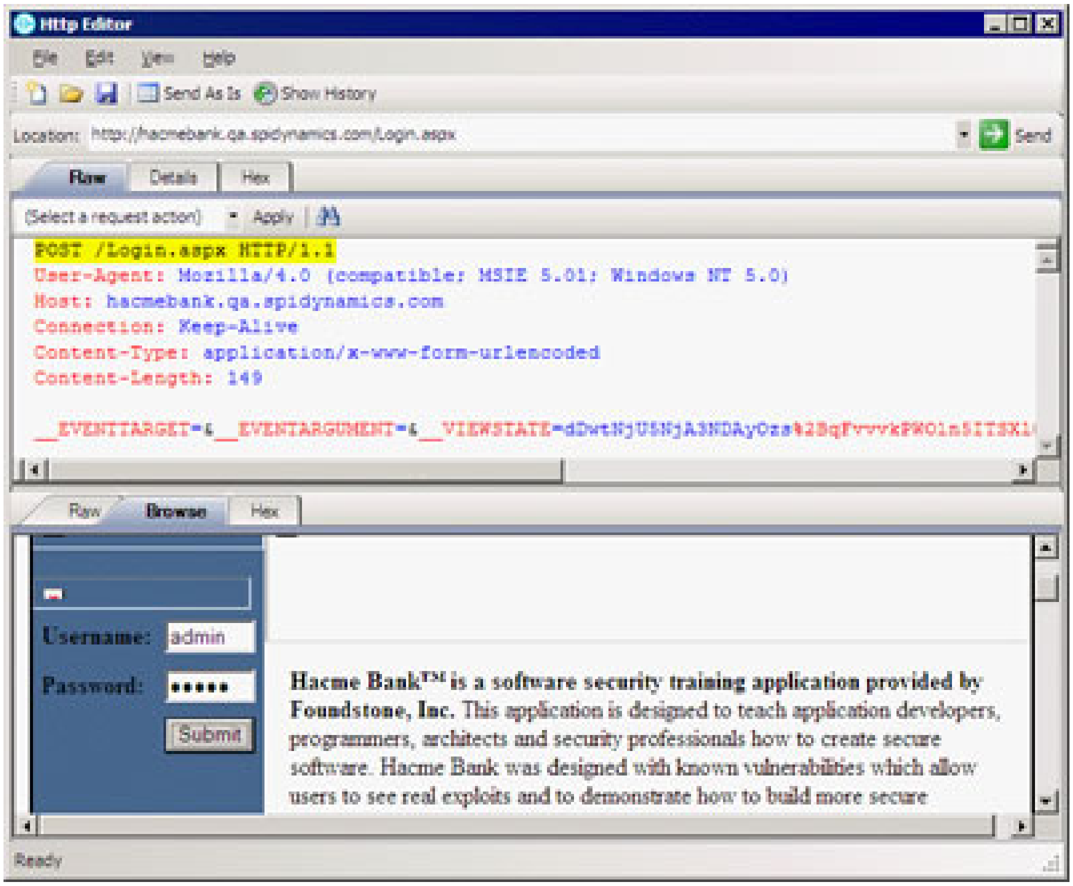
   a. In the **Location** field, enter a URL and click **Send**.

      The application returns a logon form.

   b. In the **Response** pane, click the **Browser** tab.

   c. On the rendered page, enter a user name and password, and then click **Submit**.

      The HTTP Editor formats the request (which uses the POST method to the Login.aspx URL) and displays it in the Request Viewer pane, as illustrated below.



Location: http://hacmebank.qa.spidynamics.com/Login.aspx

```
POST /Login.aspx HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: hacmebank.qa.spidynamics.com
Connection: Keep-Alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 149

__EVENTTARGET=&__EVENTARGUMENT=&__VIEWSTATE=dDwtNjU5NjA3NDAyOzs%2BqFvvvkPWO1n5ITSX1
```

Username: admin

Password: •••••

Hacme Bank™ is a software security training application provided by Foundstone, Inc. This application is designed to teach application developers, programmers, architects and security professionals how to create secure software. Hacme Bank was designed with known vulnerabilities which allow users to see real exploits and to demonstrate how to build more secure

   d. Click **Send** to send the formatted response (including the user name and password) to the server.

5. To save a request, select **File > Save Requests**.

See Also

"Searching the Request or Response" on the next page

Searching the Request or Response

To search for text in the request or response:

1. Click 🔍 in either the Request Viewer or Response Viewer pane.
2. Using either the Find in Request or Find in Response window, type or select a string or regular expression.
3. If using a regular expression as the search string, select the **Regex** check box.
4. Click **Find**.

Settings

To modify the HTTP Editor settings, click **Edit > Settings**, select one of the following tabs, make your changes, and click **OK**:

- Options
- Authentication
- Proxy

The settings on each tab are described in the following sections.

Options Tab

The **Request Group** includes the following options:

- **Send As Is** - If you select this option, the HTTP Editor will not modify the request, regardless of any other settings you may select. This allows you to send a purposely malformed message. **Authentication** and **Proxy** settings are disabled when using this option.

  > **Note:** You can manually edit the request to go through a proxy, but most standard HTTP proxy servers cannot process non-compliant HTTP requests.

- **Manipulate Request** - If you select this option, the HTTP Editor will modify requests to accommodate the following parameters:
  - **Apply State** - If your application uses cookies, URL rewriting, or post data techniques to maintain state within a session, the HTTP Editor will attempt to identify the method and modify the response accordingly.

  - **Apply Proxy** - If you select this option, the HTTP Editor will modify the request according to the proxy settings you specify.

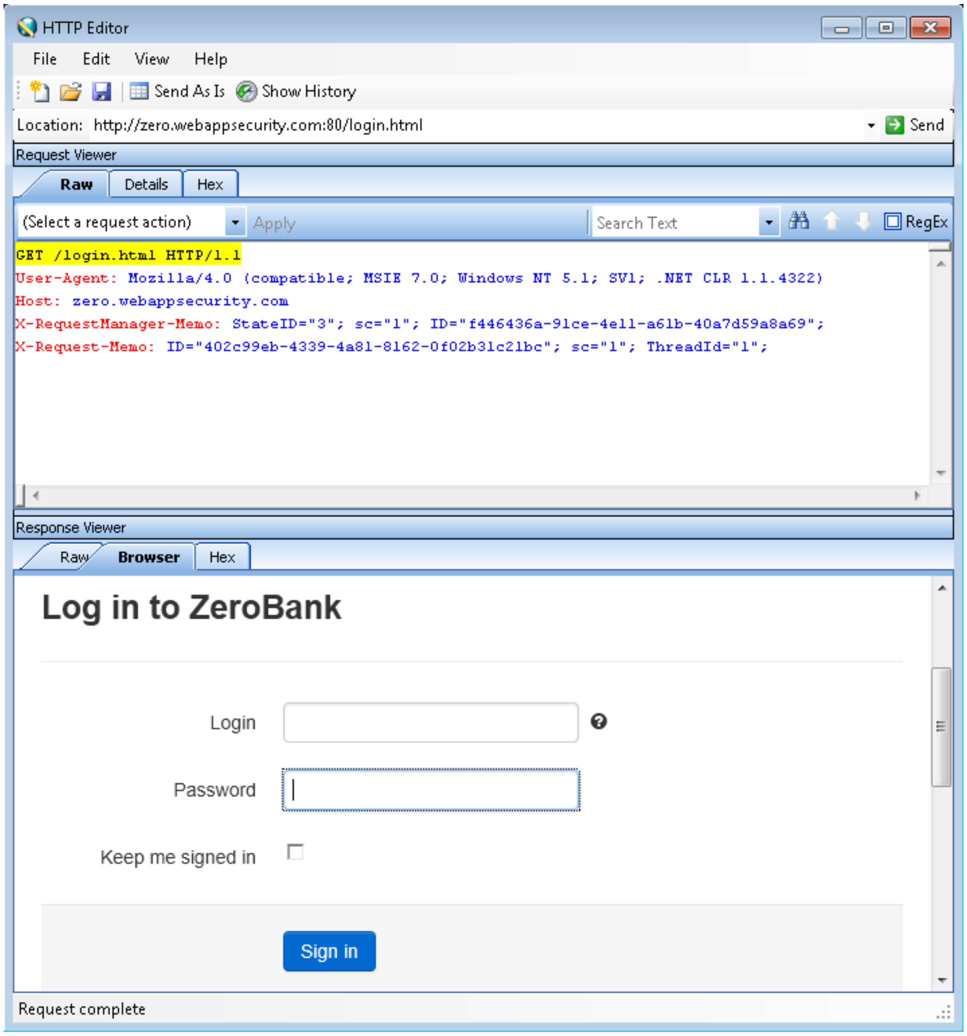
  - **Apply Filter** - This option appears only when you invoke the HTTP Editor while using Fortify WebInspect and a scan tab has focus (that is, after opening or while conducting a scan). If this option is selected, the HTTP Editor applies the Filters settings from Fortify WebInspect's Current

Scan Settings to add search-and-replace rules for HTTP requests and responses. Note that changing the Current Scan Settings before invoking the HTTP Editor has no effect; the HTTP Editor will use the settings that were in effect when the scan began.

- **Apply Header** - This option appears only when you invoke the HTTP Editor while using Fortify WebInspect and a scan tab has focus (that is, after opening or while conducting a scan). If this option is selected, the HTTP Editor applies the Cookies/Headers settings for Fortify WebInspect's Current Scan Settings for HTTP requests. Note that changing the Current Scan Settings before invoking the HTTP Editor has no effect; the HTTP Editor will use the settings that were in effect when the scan began.

In the **Navigation group**, select **None, Interactive,** or **Browser Mode**.

You can view the server's response as rendered in a browser by selecting the **Browser** tab in the Response Viewer (the lower pane). If the **Interactive** feature is enabled, you can prepare your next HTTP request using the HTML or JavaScript controls rendered in the browser.



Location: http://zero.webappsecurity.com:80/login.html

```
GET /login.html HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Host: zero.webappsecurity.com
X-RequestManager-Memo: StateID="3"; sc="1"; ID="f446436a-91ce-4e11-a61b-40a7d59a8a69";
X-Request-Memo: ID="402c99eb-4339-4a81-8162-0f02b31c21bc"; sc="1"; ThreadId="1";
```

Log in to ZeroBank

Login

Password

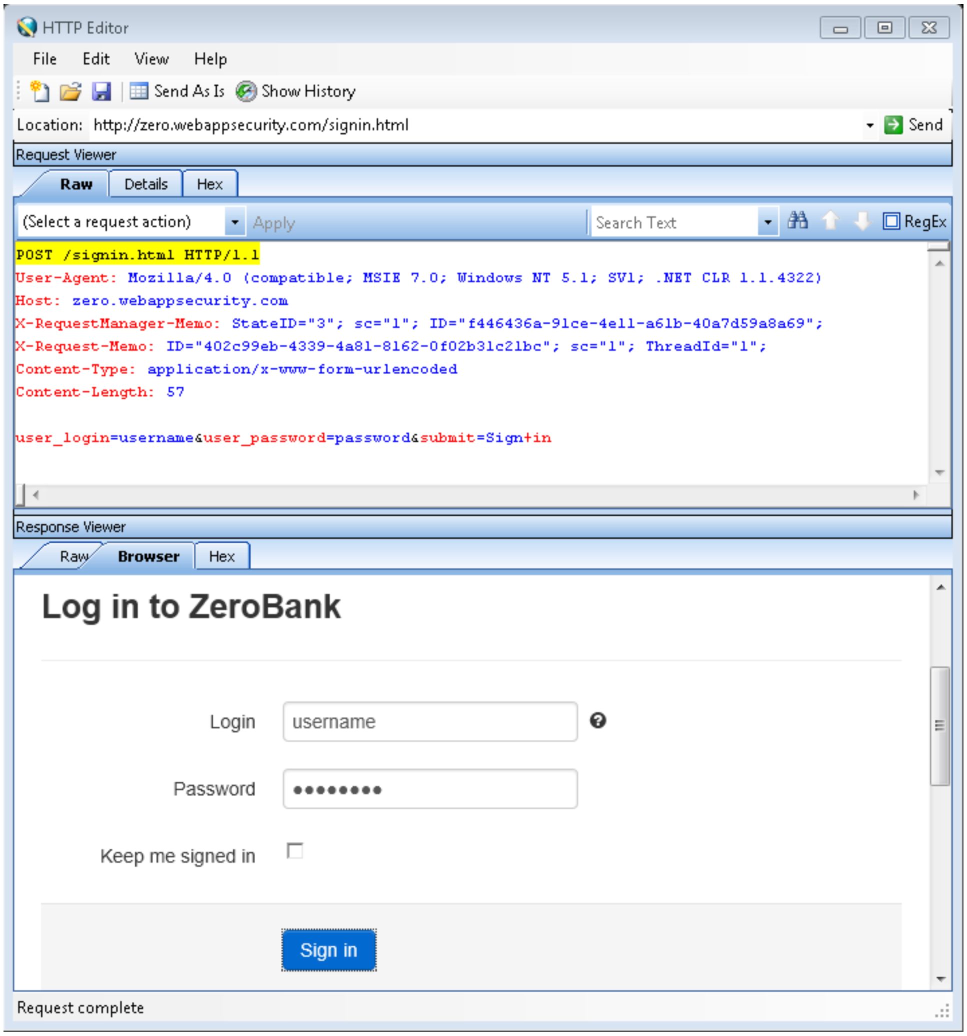
Keep me signed in □

Sign in

Request complete

For example, using the logon page at http://zero.webappsecurity.com:80/login.html (shown above), you could enter a **Login** name ("username") and **Password** ("password"), and then click **Sign in**. The

HTTP Editor formats the request (which uses the POST method to the signin.html resource) and displays it in the Request Viewer, as illustrated below. You could then edit the logon message (if required) or simply send it to the server by clicking **Send**.



Location: http://zero.webappsecurity.com/signin.html

```
POST /signin.html HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)
Host: zero.webappsecurity.com
X-RequestManager-Memo: StateID="3"; sc="1"; ID="f446436a-91ce-4e11-a61b-40a7d59a8a69";
X-Request-Memo: ID="402c99eb-4339-4a81-8162-0f02b31c21bc"; sc="1"; ThreadId="1";
Content-Type: application/x-www-form-urlencoded
Content-Length: 57

user_login=username&user_password=password&submit=Sign+in
```

Login: username

Password: ••••••••

Keep me signed in □

If you select the **Browser Mode** option, then **Interactive** mode is enabled, but the HTTP Editor will send the request immediately, without first placing it in the Request Viewer and allowing you to edit it.

Select the **Enable Active Content** check box to allow execution of JavaScript and other dynamic content in all browser windows.

Most web pages contain information that tells the browser which character set to use. This is accomplished by using the Content-Type response header (or a META tag with an HTTP-EQUIV attribute) in the HEAD section of the HTML document. For pages that do not announce their character set, you can specify which character set the HTTP Editor should use. In the **Advanced HTTP Parsing** group, select the **Assumed 'charset' Encoding**.

Authentication Tab

If authentication is required, select a type from the **Authentication** list. After selecting an authentication method, enter a user name and password. The authentication methods are:

- **Automatic** - If you select this option, the HTTP Editor will determine the authentication method used by your server.
- **HTTP Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

  The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.
- **NTLM** - NTLM (NT LanMan) is an authentication process used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network.

  Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and the scanner has to pass through a proxy server to submit its requests to the web server, the scanner may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

After selecting an authentication method, enter a **User name** and **Password**. To prevent typographical errors, you must re-enter the password in the **Confirm Password** field.

Proxy Tab

Use these settings to access the HTTP Editor through a proxy server.

- **Direct Connection (proxy disabled)** - Select this option if you are not using a proxy server.
- **Auto detect proxy settings** - Select this option to use the Web Proxy Autodiscovery Protocol (WPAD) to locate and use a proxy autoconfig file to configure the browser's web proxy settings.
- **Use System proxy settings** - Select this option to import your proxy server information from the local machine.
- **Use Firefox proxy settings** - Select this option to import your proxy server information from Firefox.

> **Note:** Using browser proxy settings does not guarantee that you will access the Internet through

a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.

- **Explicitly configure proxy** - Select this option to access the Internet through a proxy server, and then enter the requested information:

  a. In the **Server** field, type the URL or IP address of your proxy server, followed (in the **Port** field) by the port number (for example, 8080).

  b. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.

  c. If authentication is required, select a type from the **Authentication** list:

     - **Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

       The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.

     - **NTLM** - NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

       Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and the scanner has to pass through a proxy server to submit its requests to the web server, the scanner may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

     - **Kerberos** - Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

     - **Digest** - The Windows Server operating system implements the Digest authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using Digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

     - **Automatic** - Allow HTTP Editor to determine the correct authentication type.

> **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

- ○ **Negotiate** - The Negotiate authentication protocol begins with the option to negotiate for an authentication protocol. When the client requests access to a service, the server replies with a list of authentication protocols that it can support and an authentication challenge based on the protocol that is its first choice. For example, the server might list Kerberos and NTLM, and send a Kerberos challenge. The client examines the contents of the reply and checks to determine whether it supports any of the specified protocols. If the client supports the preferred protocol, authentication proceeds. If the client does not support the preferred protocol, but does support one of the other protocols listed by the server, the client lets the server know which authentication protocol it supports, and the authentication proceeds. If the client does not support any of the listed protocols, the authentication exchange fails.

d. If your proxy server requires authentication, enter the qualifying **User name** and **Password**.

e. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** field. Use commas to separate entries.

- **Specify Alternative Proxy for HTTPS**

  For proxy servers accepting HTTPS connections, select the **Specify Alternative Proxy for HTTPS** check box and provide the requested information.

Regular Expressions

Special characters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used. Another recommended resource is the Regular Expression Library.

Character	Description
\	Marks the next character as special. /n/ matches the character " n ". The sequence /\n/ matches a line feed or newline character.
^	Matches the beginning of input or line. Also used with character classes as a negation character. For example, to exclude everything in the content directory except /content/en and /content/ca, use: /content/[^(en|ca)].*/.* . Also see \S \D \W.
$	Matches the end of input or line.
*	Matches the preceding character zero or more times. /zo*/ matches either " z " or "zoo."
+	Matches the preceding character one or more times. /zo+/ matches "zoo" but not "z."

Character	Description
?	Matches the preceding character zero or one time. /a?ve?/ matches the "ve" in "never."
.	Matches any single character except a newline character.
[xyz]	A character set. Matches any one of the enclosed characters. /[abc]/ matches the "a" in "plain."
\b	Matches a word boundary, such as a space. /ea*r\b/ matches the "er" in "never early."
\B	Matches a non-word boundary. /ea*r\B/ matches the "ear" in "never early."
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a non-digit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a line feed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v].
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v].
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any non-word character. Equivalent to [^A-Za-z0-9_].

See Also

"Regular Expression Extensions" below

Regular Expression Extensions

Micro Focus engineers have developed and implemented extensions to the normal regular expression syntax. When building a regular expression, you can use the following tags and operators.

Regular Expression Tags

- [HEADERS]
- [COOKIES]
- [STATUSLINE]

- [STATUSCODE]
- [STATUSDESCRIPTION]
- [ALL]
- [BODY]
- [SETCOOKIES]
- [METHOD]
- [REQUESTLINE]
- [VERSION]
- [POSTDATA]
- [URI]

Regular Expression Operators

- AND
- OR
- NOT
- [ ]
- ( )

Examples

- To detect a response in which (a) the status line contains a status code of "200" and (b) the phrase "logged out" appears anywhere in the message body, use the following regular expression:

  [STATUSCODE]200 AND [BODY]logged\sout
- To detect a response indicating that the requested resource resides temporarily under a different URI (redirection) and having a reference to the path "/Login.asp" anywhere in the response, use the following:

  [STATUSCODE]302 AND [ALL]Login.asp
- To detect a response containing either (a) a status code of "200" and the phrase "logged out" or "session expired" anywhere in the body, or (b) a status code of "302" and a reference to the path "/Login.asp" anywhere in the response, use the following regular expression:

  ([STATUSCODE]200 AND [BODY]logged\sout OR [BODY]session\sexpired) OR ([STATUSCODE]302 AND [ALL]Login.asp)

  Note that you must include a space (ASCII 32) before and after an "open" or "close" parenthesis; otherwise, the parenthesis will be erroneously considered as part of the regular expression.
- To detect a redirection response where "login.aspx" appears anywhere in the redirection Location header, use the following regular expression:

  [STATUSCODE]302 AND [HEADERS]Location:\slogin.aspx
- To detect a response containing a specific string (such as "Please Authenticate") in the Reason-Phrase portion of the status line, use the following regular expression:

```
[STATUSDESCRIPTION]Please\sAuthenticate
```

See Also

"Regular Expressions" on page 51

Chapter 6: About the Log Viewer Tool (Fortify WebInspect Only)

Use the Log Viewer to inspect the various logs maintained by Fortify WebInspect. This feature is used mainly by the Fortify Customer Support group to investigate reported incidents.

To view log files:

1. Click the **Tools > Log Viewer**.

   If you open the Log Viewer when a tab containing a scan has focus, the program assumes you want to view logs for that scan. Go to Step 4.
2. Click **Open Scan**.
3. On the Open Scan window, select the scan whose logs you want to view and click **Open**. To open scans in a different database, click **Change Database**.
4. Select a log from the **Log Type** list. The available types depend on the logging level that was selected for the scan (in Fortify WebInspect's Application settings).
5. To locate text within the log, click **Find** on the toolbar

   - or -

   Select **Edit > Find**.
6. To save a log file, click **Export** on the toolbar

   - or -

   Select **File > Export Logs**.
7. To view logs that are not related to a specific scan, click **WebInspect Logs** (on the toolbar).

Chapter 7: About the Policy Manager Tool

A policy is a collection of audit engines and attack agents that Fortify WebInspect uses when auditing or crawling your web application. Each component has a specific task, such as testing for cross-site scripting susceptibility, building the site tree, probing for known server vulnerabilities, etc. These components are organized into the following groups:

- Audit Engines
- Audit Options
- General Application Testing
- General Text Searching
- Third-Party Web Applications
- Web Frameworks/Languages
- Web Servers
- Web Site Discovery
- Custom Agents
- Custom Checks

All these components (except for the Audit Engines) are known collectively as attack groups. Each attack group contains subgroups of individual modules (called attack agents) that check your website for vulnerabilities.

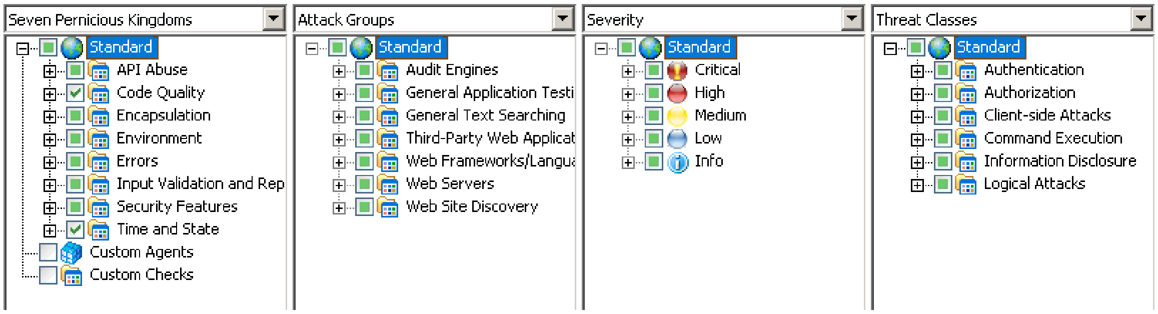
Fortify WebInspect contains several prepackaged policies designed to accommodate the requirements of most users. All policies contain all possible audit engines and agents, but each policy has a different subset of these components enabled. You edit a policy by enabling or disabling audit engines and/or individual attack agents (or groups of agents). You create a policy by editing an existing policy and saving it with a new name.

Views

The Policy Manager has two different views—**Standard** and **Search**—which are selectable from the **View** menu by clicking icons in the toolbar.

Standard View

This view displays, by default, a list of checks categorized by Seven Pernicious Kingdoms. Alternatively, a drop-down list allows you to display checks by Attack Groups, Severity, and Threat Class (according to classifications established by the Web Application Security Consortium).



Seven Pernicious Kingdoms
- Standard
  - API Abuse
  - Code Quality
  - Encapsulation
  - Environment
  - Errors
  - Input Validation and Rep
  - Security Features
  - Time and State
  - Custom Agents
  - Custom Checks

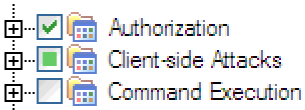
Attack Groups
- Standard
  - Audit Engines
  - General Application Testi
  - General Text Searching
  - Third-Party Web Applicat
  - Web Frameworks/Langua
  - Web Servers
  - Web Site Discovery

Severity
- Standard
  - Critical
  - High
  - Medium
  - Low
  - Info

Threat Classes
- Standard
  - Authentication
  - Authorization
  - Client-side Attacks
  - Command Execution
  - Information Disclosure
  - Logical Attacks

You enable or disable a component by selecting or clearing its associated check box.

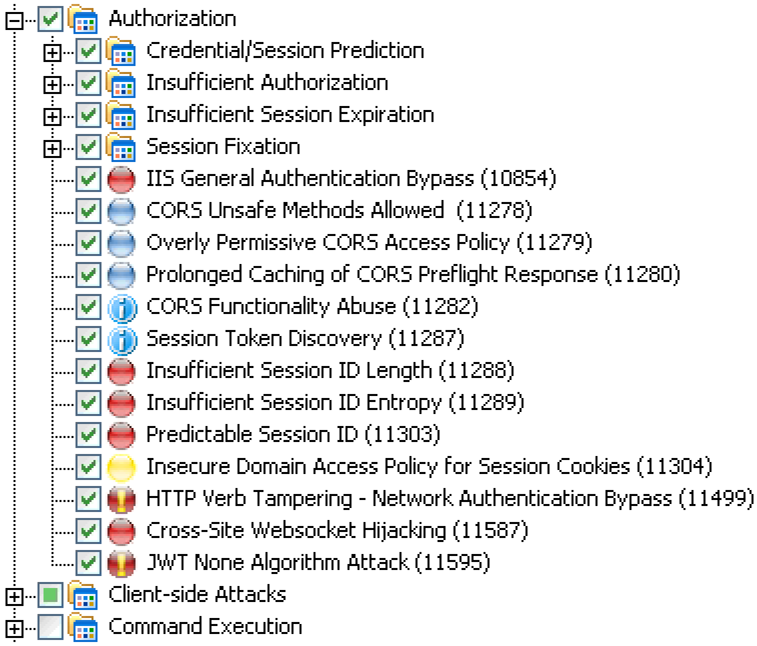
The check box next to an unexpanded node indicates the "selected" status of the objects within the node.



☑ Authorization
🟩 Client-side Attacks
☐ Command Execution

- A **check** means all objects are selected.
- A **green square** means some objects are selected.
- An **empty box** means no objects are selected.

Click the plus sign ⊞ to expand a node.



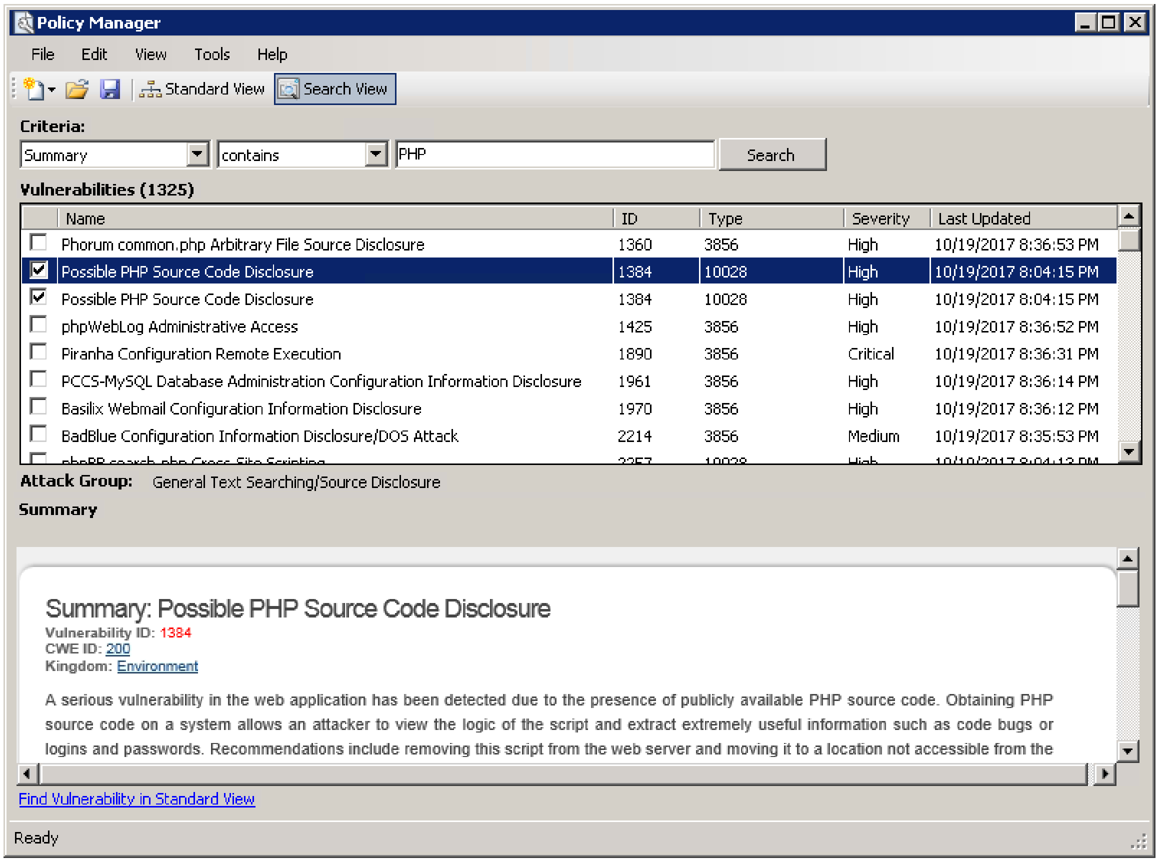
- ☑ Authorization
  - ☑ Credential/Session Prediction
  - ☑ Insufficient Authorization
  - ☑ Insufficient Session Expiration
  - ☑ Session Fixation
  - ☑ IIS General Authentication Bypass (10854)
  - ☑ CORS Unsafe Methods Allowed (11278)
  - ☑ Overly Permissive CORS Access Policy (11279)
  - ☑ Prolonged Caching of CORS Preflight Response (11280)
  - ☑ CORS Functionality Abuse (11282)
  - ☑ Session Token Discovery (11287)
  - ☑ Insufficient Session ID Length (11288)
  - ☑ Insufficient Session ID Entropy (11289)
  - ☑ Predictable Session ID (11303)
  - ☑ Insecure Domain Access Policy for Session Cookies (11304)
  - ☑ HTTP Verb Tampering - Network Authentication Bypass (11499)
  - ☑ Cross-Site Websocket Hijacking (11587)
  - ☑ JWT None Algorithm Attack (11595)
- 🟩 Client-side Attacks
- ☐ Command Execution

Search View

This view allows you to locate attack agents based on the attribute you select from the Criteria list:

- Vulnerability ID
- Vulnerability Name
- Engine Type
- Last Updated
- CWE ID
- Kingdom
- Summary
- Implication
- Execution
- Fix
- Reference Info

This feature is used most often to identify checks that you want to disable. For example, if you are scanning an application that does not contain PHP scripting, you could search summary fields for "PHP." When the Policy Manager lists the attack agents that match your search criteria, you could disable an agent by clearing its associated check box. Then, you can either save the modified policy (making the policy changes permanent) or simply apply the modified policy to the current scan.


Policy Manager
File Edit View Tools Help
Standard View
Search View
Search
Find Vulnerability in Standard View

Criteria: Summary | contains | PHP

Vulnerabilities (1325)

	Name	ID	Type	Severity	Last Updated
☐	Phorum common.php Arbitrary File Source Disclosure	1360	3856	High	10/19/2017 8:36:53 PM
☑	Possible PHP Source Code Disclosure	1384	10028	High	10/19/2017 8:04:15 PM
☑	Possible PHP Source Code Disclosure	1384	10028	High	10/19/2017 8:04:15 PM
☐	phpWebLog Administrative Access	1425	3856	High	10/19/2017 8:36:52 PM
☐	Piranha Configuration Remote Execution	1890	3856	Critical	10/19/2017 8:36:31 PM
☐	PCCS-MySQL Database Administration Configuration Information Disclosure	1961	3856	High	10/19/2017 8:36:14 PM
☐	Basilix Webmail Configuration Information Disclosure	1970	3856	High	10/19/2017 8:36:12 PM
☐	BadBlue Configuration Information Disclosure/DOS Attack	2214	3856	Medium	10/19/2017 8:35:53 PM
☐	phpBB search.php Cross Site Scripting	2257	10028	High	10/19/2017 8:04:13 PM

Attack Group: General Text Searching/Source Disclosure

Summary

Summary: Possible PHP Source Code Disclosure

Vulnerability ID: 1384
CWE ID: 200
Kingdom: Environment

A serious vulnerability in the web application has been detected due to the presence of publicly available PHP source code. Obtaining PHP source code on a system allows an attacker to view the logic of the script and extract extremely useful information such as code bugs or logins and passwords. Recommendations include removing this script from the web server and moving it to a location not accessible from the

Ready

See Also

"Policies" on page 79

"Creating or Editing a Policy" below

"Searching for Specific Agents" on page 69

"Creating a Custom Check" on the next page

Creating or Editing a Policy

Fortify WebInspect contains a number of prepackaged policies designed to accommodate the majority of users. You cannot permanently change these policies. However, you can open any of them as a template, modify their contents to create a custom policy, and save the customized policy under a new name. You can edit and save a custom policy without changing its name.

To edit or create a policy:

1. On the toolbar, click **Policy Manager**

   - or -

   select **Tools > Policy Manager**.

   The Policy Manager opens. By default, it loads the Standard policy.
2. To edit a policy that you previously created (that is, a custom policy), select **File > Open** and select the policy.
3. To create a policy based on one of the prepackaged policies, select **File > New** (or click the New Policy icon) and select the policy on which the new one will be modeled.
4. Disable (or enable) an attack group by clearing (or selecting) its associated check box. To disable or enable an individual agent within a group, first expand the group and then edit its check box.
5. To rename an attack group:
   a. Right-click the attack group.
   b. Choose **Rename** from the shortcut menu.
6. To add an attack group:
   a. Right-click any existing attack group.
   b. Choose **New Attack Group** from the shortcut menu.

      A highlighted entry named New Attack Group will appear.
   c. Right-click the new group and choose **Rename**.
   d. Populate the group by dragging and dropping attack agents onto it.
7. You can also create a custom check. For more information, see "Creating a Custom Check" on the next page.
8. If you select the **Auto Update** check box, Fortify WebInspect determines if any updated or new attack agents downloaded from the Micro Focus database should be enabled or disabled, based on the analysis of its sibling agents. For example, if you disable attack agents targeting Microsoft's Internet Information Server (IIS), and you select **Auto Update**, then Fortify WebInspect will not

enable any IIS-related attack agent that it downloads to your system. Conversely, any new or updated attack agents that are related to agents that are enabled in your policy will also be enabled.

> **Note:** New vulnerability checks downloaded via Smart Update are not added automatically to any custom policies you may have created.

9. Select **File > Save As**. Type a name for your custom policy in the **File name** field and then click **Save** to save the new policy in Fortify WebInspect's *.policy format. You cannot save a policy using the name of a default policy (Assault, Blank, Standard, etc.).

See Also

"Using a Custom Agent" on page 70

"Searching for Specific Agents" on page 69

"Creating a Custom Check" below

Creating a Custom Check

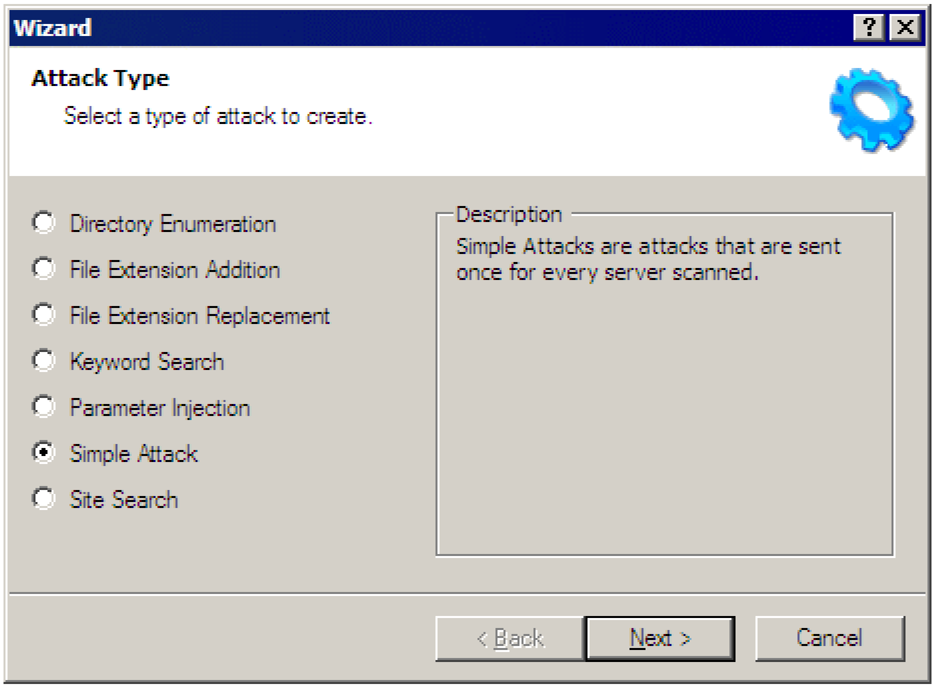
Although Fortify WebInspect rigorously inspects your entire website for real and potential security vulnerabilities, you may require a custom check to detect vulnerabilities that are unique to your application.

If you create a custom check that duplicates an attack conducted by Fortify WebInspect, your new check will not be submitted unless you disable the standard check. For example, Fortify WebInspect normally runs a directory enumeration check that searches for a backup directory with "(copy)" suffix. If you create a custom check that also searches for a backup directory with "(copy)" suffix, Fortify WebInspect will not submit it (because it has already searched for that directory) unless you disable check #11485 named Backup Directory ((copy)).

To create a custom check:

1. Open the Policy Manager.
2. To edit a policy that you previously created, select **File > Open** and select the policy.
3. To create a new policy based on a prepackaged policy, select **File > New** (or click the New Policy icon) and select the policy on which you will model a new one.
4. Make sure the **Standard view** is selected, with Seven Pernicious Kingdoms listed in the left pane.
5. Right-click on **Custom Checks** and select **New Custom Check** from the shortcut menu.

The Custom Check Wizard appears.



Simple Attacks are attacks that are sent once for every server scanned.

6. Select one of the following attack types, listed with detailed explanations and examples:

   - **Directory enumeration**

     This type of check searches for a directory of the name you specify.

     - Attack Type: Directory Enumeration
     - Attack: /directory_name/ [where directory_name is the name of the directory you want to find]
     - Signature: [STATUSCODE]3\d\d OR [STATUSCODE]2\d\d OR [STATUSCODE]40[13]

   - **File extension addition**

     This type of check searches for files with a file extension that you specify.

     During the crawl, whenever Fortify WebInspect encounters a file of any name and any extension (for example, global.asa), it sends an HTTP request for a file of the same name plus the found extension plus an extension that you specify. For example, if you specify a file extension of .backup, then when Fortify WebInspect discovers a file named global.asa, it will subsequently search for a file named global.asa.backup.

     A server would normally deny any request for the global.asa file, but if a programmer has left a backup file on the server and the file has a different extension (such as global.asa.backup), then the server might return the file (which contains the full source of the global.asa file).

     To create a custom check that searches for files with a specific added extension, enter the following in the Custom Check Wizard:

     - Attack Type: File Extension Addition
     - Attack: .ext [where ext is the file extension of files you want to locate]. You must include the leading dot or period (.)

- Signature: [STATUSCODE]200 AND ([HEADERS]Content-Type:\stext/plain OR [HEADERS]Content-Type:\sapplication/octet-stream)

- **File extension replacement**

  This type of check searches for files with a file extension that you specify.

  For example, Fortify WebInspect contains a standard check that searches for files having an extension of .old. During the crawl, whenever it encounters a file of any name and any extension (for example, startup.asp), it sends an HTTP request for a file of the same name but with an extension of .old (for example, startup.old).

  To create a custom check that searches for files with a specific extension, enter the following in the Custom Check Wizard:

  - Attack Type: File Extension Replacement
  - Attack: ext [where ext is the file extension of files you want to locate]. Do NOT include a leading dot or period (.)
  - Signature: [STATUSCODE]200 AND ([HEADERS]Content-Type:\stext/plain OR [HEADERS]Content-Type:\sapplication/octet-stream)

- **Keyword search**

  This type of check determines if a specified word or phrase (defined by a regular expression) exists anywhere in the HTTP response.

  The following example searches the HTTP response for a nine-digit number formatted as a social security number (\d = any digit).

  - Attack Type: Keyword Search
  - Attack: N/A
  - Signature: [BODY]\d\d\d-\d\d-\d\d\d\d

- **Parameter injection**

  This type of attack replaces an argument value with an attack string.

  Example:

  http://www.samplesite.com/webapp.asp?ValidParameter=ValidArgument

  will be changed to

  http://www.samplesite.com/webapp.asp?ValidParameter=AttackArgument

  There are several types of parameter injection, as follows:

  - Command Execution

    A command execution check combines strings composed of special characters with operating system-level commands. It is an attempt to make the web application execute the command using the provided string (if the application fails to check for and prohibit the input).

    The following example tests for parameter injection by providing spurious input to a program named support_page.cgi; if the HTTP response contains data that matches the regular expression, then the application is vulnerable to command execution.

    - Attack Type: Parameter Injection
    - Attack: /support_page.cgi?file_name=|id|

  - Signature: [BODY]uid= AND [BODY]gid=

- SQL Injection

  SQL injection is the act of passing SQL code into an application. These attack strings are composed of fragments of SQL syntax that will be executed on the database server if the web application uses the string when forming a SQL statement without first filtering out certain characters.

  - Attack Type: Parameter Injection
  - Attack: ' [an apostrophe]
  - Signature: [STATUSCODE]5\d\d

- Cross-Site Scripting

  This issue occurs when dynamically generated web pages display input that is not properly validated. This allows an attacker to embed malicious JavaScript into the generated page, enabling him to execute the script on the machine of any user who views the malicious page. Any site that allows users to post text messages can be vulnerable to an attack such as this.

  The following example tests for cross-site scripting in the Fusion News application:

  - Attack Type: Parameter Injection
  - Attack: /fullnews.php?id=<script>alert(document.cookie)</script>
  - Signature: [ALL]Powered\sby\sFusion\sNews And [ALL]<script>alert\(document\.cookie\)</script>

- Directory Traversal

  Directory traversal entails sending malformed URL strings to access non-public portions of the web server’s content. An attacker will try to access different files on a server by using relative hyperlinks. For example, by adding triplets of two periods and a forward slash (../) to the target URL and by varying the number of directories to traverse, an attacker might find and gain access to a system password file such as www.server.com/../../../../password.

  The following example searches for the boot.ini file:

  - Attack Type: Parameter Injection
  - Attack: /../../../../../../../../../../../boot.ini
  - Signature: [ALL]\[boot\sloader\]

- Abnormal Input

  Abnormal input attack strings are composed of characters that can cause unhandled exceptions (errors the program is not coded to handle) in web applications where unexpected input is not prohibited. Unhandled exceptions often cause servers to display error messages that disclose sensitive information about the application’s internal mechanics. Source code may even be disclosed.

  The following example sends an extraordinarily long string in an attempt to create a buffer overflow.

  - Attack Type: Parameter Injection
  - Attack: AAAAAAAAAAAA...AAAAAAAA [1000 repetitions of the letter "A"]
  - Signature: [STATUSCODE]5\d\d

- **Simple attack**

  This type of attack is sent once for every server scanned.

  The following example attempts to obtain a UNIX password file by appending the attack string to the target URL or IP address:

  - Attack Type: Simple Attack
  - Attack: /etc/passwd
  - Signature: [ALL]root: AND [ALL]:0:0

- **Site search**

  This type of attack is designed to find files commonly left on a web server. For example, check ID #279 searches for a file named log.htm.

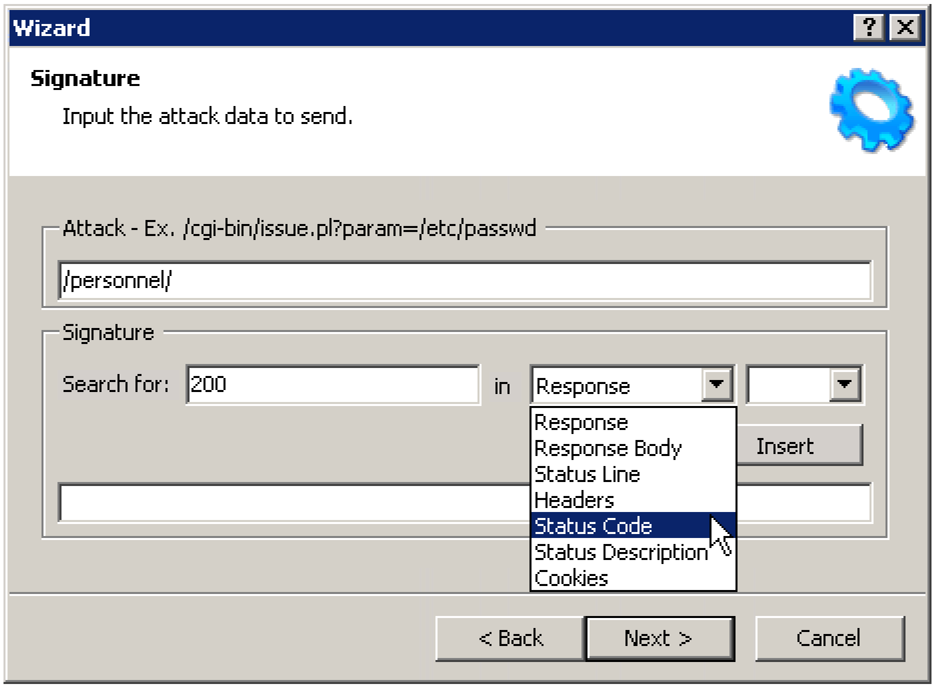
  The following example searches for a file named xanadu.html by appending the attack string to the target URL or IP address:

  - Attack Type: Site Search
  - Attack: xanadu.html
  - Signature: [STATUSCODE]2\d\d OR [STATUSCODE]40[1]

  To create a custom check that searches for a file named confidential.txt, enter the following in the Custom Check Wizard:

  - Attack Type: Site Search
  - Attack: confidential.txt
  - Signature: [STATUSCODE]2\d\d AND ([HEADERS]Content-Type:\stext/plain OR [HEADERS]Content-Type:\sapplication/octet-stream)

7. Click **Next**.
8. In the **Attack** field, enter the data you want to use for the attack.



Signature

Input the attack data to send.

Attack - Ex. /cgi-bin/issue.pl?param=/etc/passwd: /personnel/

Signature

Search for: 200 in Response

In the above example of directory enumeration, the check will search for a directory named "personnel" by appending the attack string (/personnel/) to the target URL or IP address.

9. You must specify a signature, which is simply a regular expression (that is, a special text string for describing a search pattern). When Fortify WebInspect searches the HTTP response and finds the text described by the signature, it flags the session as a vulnerability. You can use the **Search for** field and drop-down lists to help you create the regular expression, or you can type the regular expression directly into the text box at the bottom of the window.

   To use the **Search for** field:

   a. Enter the text you want to locate.

      Enter only text in the **Search for** field; do not enter a regular expression.

      In this example (searching for a directory named "personnel"), the server would return a status code of 200 if the directory exists, so enter "200" in the **Search for** field. Realistically, however, you might also accept any status code in the 200 or 300 series, or a status code of 401 or 403.

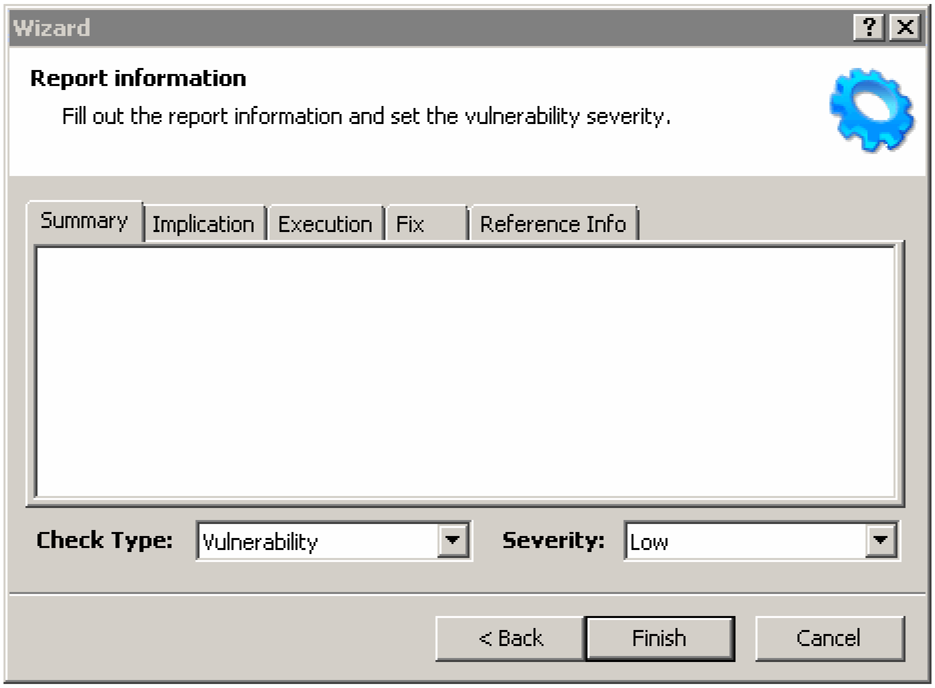
   b. Click the drop-down arrow to specify the section of the HTTP response that should be searched.

   c. (Optional) To create a complex search, click the second drop-down and select a Boolean operator (AND, OR, or NOT).

   d. Click **Insert**.

   e. (Optional) For complex searches, repeat steps a-d as needed. You can also edit or replace the regular expression that appears in the bottom text box.

10. Click **Next**.



Report information

Fill out the report information and set the vulnerability severity.

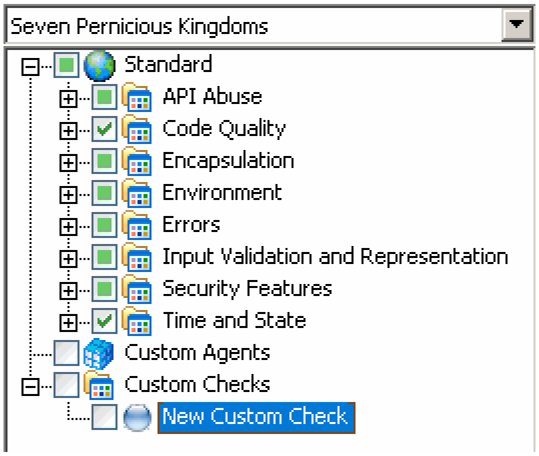
Check Type: Vulnerability

Severity: Low

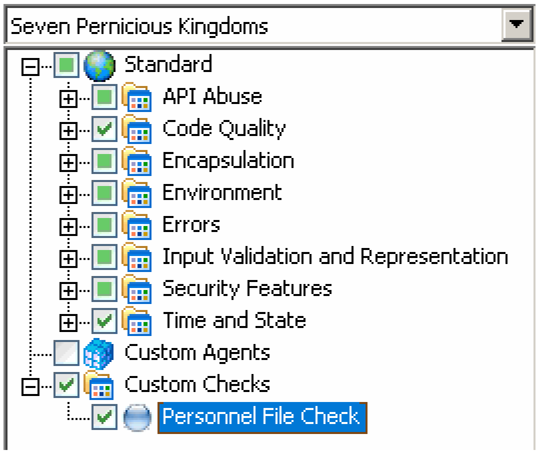
11. On the Report Information panel, click each tab and enter the text that will appear in the description.

12. Select an entry from the **Check Type** list.

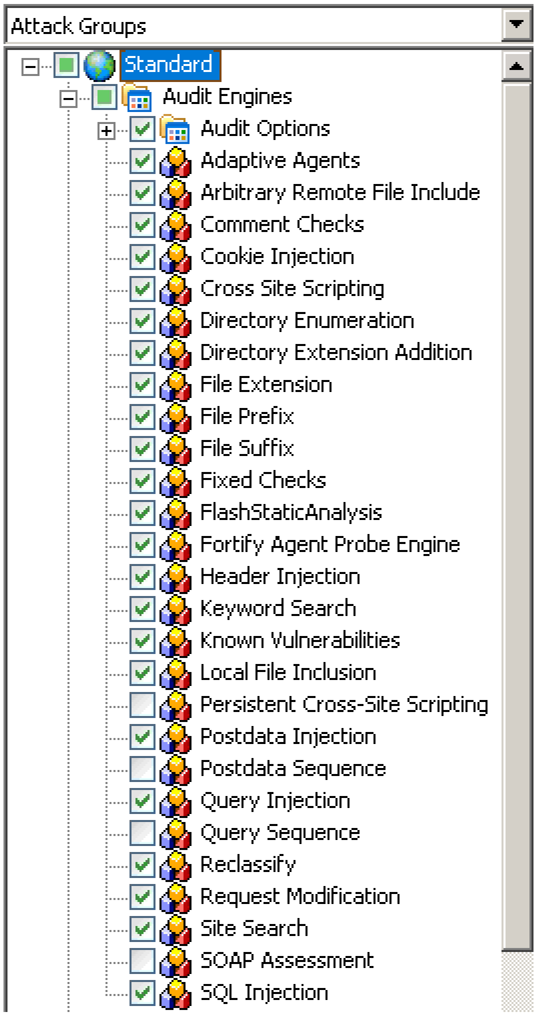
13. Select a severity level from the **Severity** list.
14. Click **Finish**.
15. Change the default name "New Custom Check" to reflect the purpose of the check.



16. Ensure that the custom check is enabled (with a check mark).



17. Select **Attack Groups** from the drop-down list, and then click ⊞ to expand the Audit Engines folder.



Attack Groups

- ⊟ 🟩 Standard
  - ⊟ 🟩 Audit Engines
    - ⊞ ☑ Audit Options
    - ☑ Adaptive Agents
    - ☑ Arbitrary Remote File Include
    - ☑ Comment Checks
    - ☑ Cookie Injection
    - ☑ Cross Site Scripting
    - ☑ Directory Enumeration
    - ☑ Directory Extension Addition
    - ☑ File Extension
    - ☑ File Prefix
    - ☑ File Suffix
    - ☑ Fixed Checks
    - ☑ FlashStaticAnalysis
    - ☑ Fortify Agent Probe Engine
    - ☑ Header Injection
    - ☑ Keyword Search
    - ☑ Known Vulnerabilities
    - ☑ Local File Inclusion
    - ☐ Persistent Cross-Site Scripting
    - ☑ Postdata Injection
    - ☐ Postdata Sequence
    - ☑ Query Injection
    - ☐ Query Sequence
    - ☑ Reclassify
    - ☑ Request Modification
    - ☑ Site Search
    - ☐ SOAP Assessment
    - ☑ SQL Injection

18. Ensure that the appropriate audit engine is enabled (with a check mark) for the type of check you created, according to the following table.

This Attack Type...	Uses This Audit Engine...
Directory Enumeration	Directory Enumeration
File Extension Addition	File Extension
File Extension Replacement	File Extension
Keyword Search	Keyword Search
Parameter Injection	Post Data Injection
Simple Attack	Fixed Checks

This Attack Type...	Uses This Audit Engine...
Site Search	Site Search

19. Select **File > Save**.
20. Enter a name for the new policy and click **Save**.

Fortify WebInspect adds all custom checks to every policy, but does not enable them. To enable the custom check in other policies, see "Creating or Editing a Policy" on page 59.

To disable a custom check:

1. Select a custom check.
2. Clear its associated check box.

To delete a custom check:

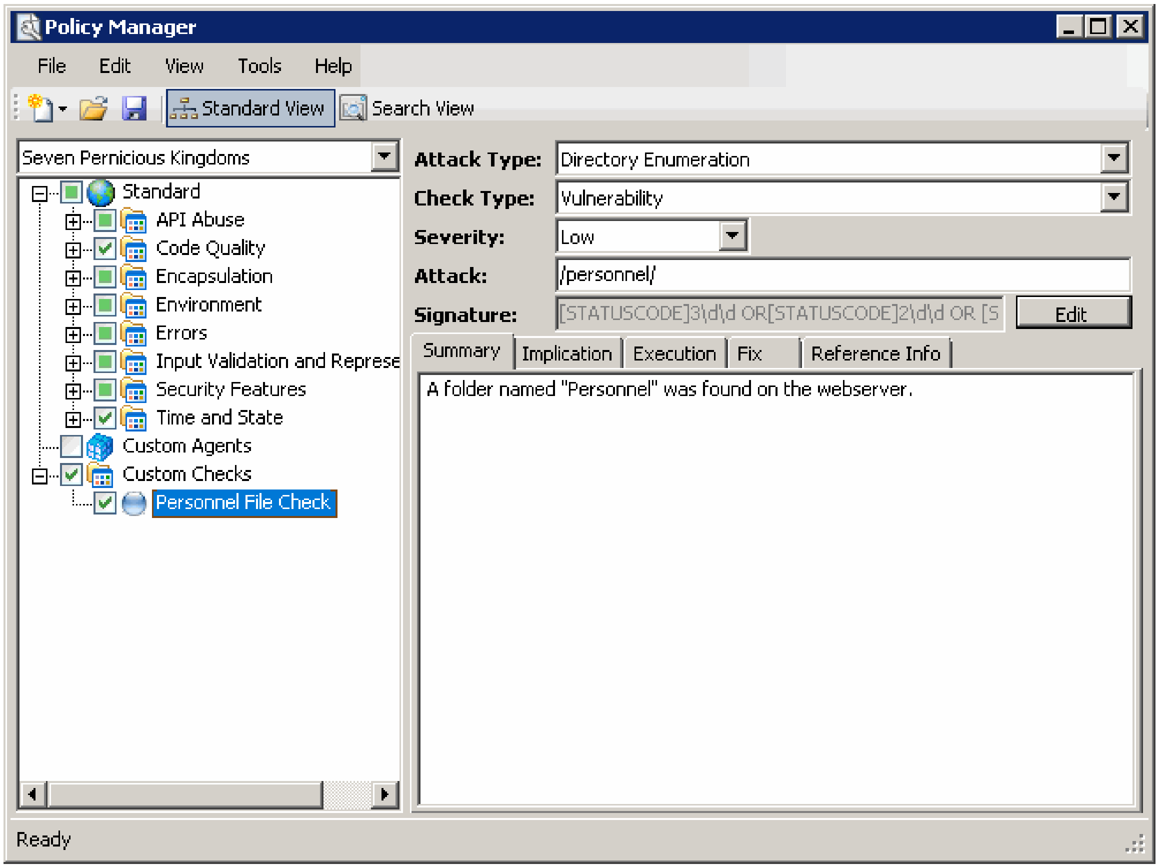
Caution! If you delete a custom check from a policy, you delete it from all policies and from the entire system.

1. Right-click a custom check.
2. Select **Delete** from the short-cut menu.

To edit a custom check:

1. Open a policy.
2. Select a custom check.

3. Using the right pane of the Policy Manager, modify the custom check properties.



Attack Type: Directory Enumeration

Check Type: Vulnerability

Severity: Low

Attack: /personnel/

Signature: [STATUSCODE]3\d\d OR[STATUSCODE]2\d\d OR [S

A folder named "Personnel" was found on the webserver.

4. Click the Save icon.

See Also

"Regular Expressions" on page 88

"Regular Expression Extensions" on page 89

Searching for Specific Agents

Use the Search view on the Policy Manager to locate specific vulnerability checks (attack agents). You can then elect to include or exclude individual agents.

To search for attack agents:

1. On the toolbar, click **Policy Manager**

   - or -

   select **Tools > Policy Manager**.
2. If you do not have a policy selected, select a policy from the *Open Policy* window and click **OK**.
3. Select **View > Search**.

The description of every attack agent contains "report fields" such as summary, implication, execution, recommendation, and fix. The Search feature allows you to locate attack agents that contain the text you specify in a selected report field.

4. From the **Criteria** list, select the report field that you want to search.
5. Choose an operator from the drop-down list (is, is greater than, is less than, contains).
6. In the text box, type the text or number you want to find.
7. Click **Search**.
8. The Policy Manager lists in the **Checks** area all attack agents that match your search criteria. An active agent has a check mark next to its name. Select (or clear) a check box to activate (or deactivate) an agent.
9. Click **Save** to save the revised policy.

Using a Custom Agent

Fortify WebInspect audit extensions are developed by software developers in your organization and published to SecureBase as custom agents that can be enabled in policies and used in conducting scans. To enable a custom agent in the Policy Manger:

1. Do one of the following:
   - To create a new policy that includes only the custom agent check, select **File > New > Blank Policy**, and go to Step 2.
   - To enable the custom agent check along with other checks in an existing policy, go to Step 2.
2. Select **Seven Pernicious Kingdoms** from the drop-down list.
3. Expand the **Custom Agents** group.
4. Select a custom agent from the list.
5. Select **File > Save**.

When conducting a scan, select the policy that includes the enabled custom agent check.

Note: If the developer republishes an extension, you must close and re-open the Policy Manager to get the revised custom agent.

Methodologies

A web application includes not only the code that creates your website, but also the architectural components necessary to make a website available and useful to the public. When considering web application security, you must account for all the components that work together to create a website, not just the visible face presented to the world at large.

Fortify WebInspect can analyze any web application, identify potential security flaws, and supply you with the latest information necessary to resolve security issues before unauthorized users are able to capitalize on them. In an ever-changing, dynamic environment like the web, having a security tool that’s always up to date is an absolute necessity. With this in mind, Micro Focus's design team engineered the software to automatically update its built-in knowledgebase of known successful hacking methodologies every time it’s used. The software will then emulate these methodologies against the applications to be tested. This knowledgebase is gathered from Micro Focus security experts, as well as a wide variety of leading third-party security organizations and analysts.

When new methods of attack are discovered, Micro Focus is ready with same-day upgrades to its SecureBase™ vulnerabilities database. Following is a list of the key methodologies that Fortify WebInspect employs when assessing the security vulnerabilities of your web application.

Parameter Manipulation

Parameter manipulation involves tampering with URL parameters to retrieve information that would otherwise be unavailable to the user. Parameter manipulation modifies, adds or removes parameter names and/or arguments. Basically, any input can be modified. Parameter manipulation attacks can be used to achieve a number of objectives, including disclosure of files above the web root, extraction of information from a database and execution of arbitrary operating-system level commands. This is applied to:

- **Query strings**. Web applications often use query strings as a simple method of passing data from the client and the server. Query strings are a way to add data calls to a hyperlink, and then retrieve that information on the linked page when it is displayed. By manipulating query strings, an attacker can easily steal information from a database, learn details about the architecture of your web application, or possibly execute commands on your web server. When conducting an audit, Fortify WebInspect implements advanced query string manipulation to ascertain the feasibility of command execution on your server(s), and determines the vulnerability of your web applications to query string manipulation.
- **Post data**. Since manipulating a query string is as easy as typing text in the address bar of a browser, many web applications rely on the POST method coupled with the use of forms rather than GET to pass data between pages. Since browsers normally don't display POST data, some programmers are lulled into thinking that it is difficult or impossible to change the data, when in fact the opposite is true. Fortify WebInspect determines your application's susceptibility to attacks that rely on the POST method of parameter manipulation.
- **Headers**. Both HTTP requests and responses use headers to deliver information about the HTTP message. A developer may not consider HTTP headers as areas of input, even though many web applications will log headers such as the "referrer" or "user-agent" to a database for traffic statistics. Fortify WebInspect intercepts header information, and attempts to pass different parameter values during an audit.
- **Cookies**. Many web applications use cookies to save information (for example, user ID's and timestamps) on the client's machine. By changing these values, or “poisoning” the cookie, malicious users can gain access to the accounts and information of other users. As well, attackers can also steal a user's cookie and gain direct access to the user's account, bypassing the need to enter an ID and password or other form of authentication. Fortify WebInspect lists all cookies discovered during a scan, and attempts to change their parameters during an audit.

Parameter manipulation can be divided into several subcategories, as described in the following sections.

Parameter Injection

Parameter injection attacks replace an argument value with an attack string.

Example:

http://www.site.com/webapp.asp?ValidParameter=ValidArgument will be changed to
http://www.site.com/webapp.asp?ValidParameter=AttackString

These attempts to manipulate parameters associated with a URL are usually directed to the following areas:

Command Execution

Command execution attack strings are composed of special characters combined with operating system-level commands that will be run if the web application uses the string in a call to an operating system command without first parsing out the special characters.

Example: ;id;

Fortify WebInspect submits harmless commands, such as the ID command, to ascertain the feasibility of commands being inserted by an attacker and then executed.

SQL Injection

SQL injection is the act of passing SQL code not intended by the developer into an application. These attack strings are composed of fragments of SQL syntax that will be executed on the database server if the web application uses the string when forming a SQL statement without first parsing out certain characters.

Example: '+(SELECT TOP 1 name FROM sysobjects WHERE 1=1)+'

Problems can arise when a developer does not protect against potentially malicious input such as an apostrophe (`'`), which could close the SQL string and give the user unintended system and application access.

Cross-Site Scripting

This issue occurs when dynamically generated web pages display input that is not properly validated. This allows an attacker to embed malicious JavaScript into the generated page, enabling him to execute the script on the machine of any user who views the malicious page. Any site that allows users to post text messages can be vulnerable to an attack such as this. This vulnerability is commonly seen on the following:

- Search engines that repeat the search keyword that was entered
- Error messages that repeat the string that contained the error
- Forms that are filled out where the values are later presented to the user
- Web message boards that allow users to post their own messages.

An attacker who uses cross-site scripting successfully might compromise confidential information, manipulate or steal cookies, create requests that can be mistaken for those of a valid user, or execute malicious code on the user systems.

Abnormal Input

Abnormal input attack strings are composed of characters that can cause unhandled exceptions (errors the program is not coded to handle) in web applications where unexpected input is not parsed out. Unhandled exceptions often cause error messages to be displayed that disclose sensitive information about the application’s internal mechanics. Source code may even be disclosed.

Example: %00

Parameter Overflow

Parameter overflow attacks supply web applications with extremely large amounts of data in the forms of parameter or cookie header arguments or parameter names. If a web application is programmed in such a manner that it cannot appropriately handle unexpected and extremely large amounts of data, it may be possible to execute arbitrary operating system-level code or cause a denial-of-service condition.

Buffer Overflow

Buffer overflow attacks can be used to execute arbitrary operating system commands. Fortify WebInspect determines whether or not you are vulnerable to buffer overflow attacks, and provides details for remedying any buffer overflow vulnerabilities.

Example:

http://www.site.com/webapp.asp?ValidParameter=ValidArgument

will be changed to

http://www.site.com/webapp.asp?XXXXXXXXXXXXXXXXX[several thousand more characters]XXX=ValidArgument

and also to

http://www.site.com/webapp.asp?ValidParameter=XXXXXXXXXXXXXXXXX[several thousand more characters]XXX

Parameter Addition

Parameter addition attacks insert new parameters into an HTTP request (such as admin=true) in an attempt to gain access to restricted or undocumented application features, and to manipulate internal application settings.

Application Debug/Backdoor Mode Parameters

Application debug/backdoor mode parameters are often undocumented application features that are added by programmers in order to assist with quality assurance. Access to debug and backdoor modes can lead to disclosure of sensitive information about the internal mechanics of the web application or even administrative control.

Example:

http://www.site.com/webapp.asp?ValidParameter=ValidArgument&debug=true

Path Manipulation

Path manipulation attacks construct or modify the Request-URI section of the HTTP request in order to gain access to files above the web root, bypass authorization settings, display directory listings or display file source. Each of the following is a method of path manipulation.

Path Truncation

Path truncation attacks are requests for known directories without file names. This may cause directory listings to be displayed. Fortify WebInspect truncates paths, looking for directory listings or unusual errors within each truncation.

Example:

If a link consists of 'http://www.site.com/folder1/folder2/file.asp' truncating the path to look for 'http://www.site.com/folder1/folder2/' and 'http://www.site.com/folder1/'. will cause the web server to reveal directory contents or to cause unhandled exceptions.

Character Encoding

Character encoding attacks substitute encoded equivalents of characters in a request for a known resource. If the web application performs a string comparison for authorization or processing purposes using the encoded URI without first parsing the encoded characters, authorization settings may be defeated or source code may be disclosed. Fortify WebInspect submits various encoded character strings to ascertain whether your web application properly parses special characters. The following elements are included when Fortify WebInspect performs character encoding tests.

- Unicode: The Unicode Worldwide Character Standard includes letters, digits, diacritics, punctuation marks, and technical symbols for all the world's principal written languages, using a uniform encoding scheme. Fortify WebInspect submits strings that have been converted to their Unicode equivalent, and attempts to gain unauthorized authentication credentials through this manipulation.
- Hexadecimal coding: This involves replacing characters with their hexadecimal equivalent. Fortify WebInspect submits hex-encoded strings, and attempts to gain unauthorized authentication credentials through this manipulation.

MS-DOS 8.3 Short Filename

MS-DOS 8.3 short filename attacks convert the file names to the MS-DOS 8.3 format (1 to 8 characters, as opposed to the 255 characters allowed for file names by more recent versions of Windows). If the web application performs a string comparison for authorization or processing purposes using the MS-DOS 8.3 filename without first converting it to its FAT32/NTFS equivalent, this may defeat authorization settings or cause source code to be disclosed.

Example: longfilename.asp would become longfi~1.asp

Directory Traversal

Directory traversal attacks are expressions in the URI that will cause the web server to display the contents of files above the web root if the web application uses the string to specify a file location without first completely parsing out traversal characters.

Example: ../../../../../boot.ini

Character Stripping

Character stripping attacks add special characters to a URI that the server or application may parse out. If the server or application uses the URI in a string comparison for authorization or request processing without first stripping out the special characters, authorization settings may be defeated and source code may be disclosed.

Character Append

Character append attacks add a special character to the end of a file or directory name.

Example: file.asp would become file.asp%00

Site Search

This can be considered the information-gathering stage, emulating an intruder's attempt to learn as much as possible about your web application before launching an attack. Site search is used to locate resources such as documents, applications and directories on the server that are not intended to be viewed by web users. Disclosure of such resources can result in the disclosure of confidential data, information about internal server and application configurations and settings, administrative access to the site, and information and application source code. Fortify WebInspect determines the availability of the following items, among others, to users of your web application.

- Test and sample files: These often contain information that can be used to implement an attack. For example, authenticated test scripts that have been left on the server could provide an attacker with the location of sensitive areas of your site.
- Administrative interfaces: These are applications that network administrators often place on a network to conduct remote maintenance.
- Application data: This can be information in a database or data passed from page to page via another method.
- Program dumps: Programs often leave a dump file on the server when they terminate prematurely. Attackers will often break an application through various methods and then retrieve important information from a dump file.
- Application logs: Several software applications leave default application logs that detail the installation of the product. Application logs can reveal important information about the architecture of your web application, including the location of hidden areas.
- Installation documentation: Certain software packages place comprising information in default installation documentation that is left available on the server.
- Backup files: Network administrators and developers often leave backup files and scripts on the web server. These files commonly contain information that can be used to breach a site's security. Backup file search involves replacing extensions on files, and then looking for older or backup versions stored on the site. For example, an attacker who finds hi.asp might search for hi.old and hi.back, and retrieve the script’s source code.
- Site statistics pages: These can be used to determine information about who is visiting your site. However, it can also reveal information that an attacker can use in formulating an attack, such as the location of other areas of your site.

Application Mapping

Fortify WebInspect exposes and follows all known (and unknown) links located on your site. This creates a baseline for vulnerability checking and application testing.

Crawl

One of the most important elements of discovering the security vulnerabilities of your web application is in mapping its internal structure. A crawl completely maps a site's tree structure. In essence, a crawl runs until no more links on the URL can be followed.

Automatic Form-Filling

Fortify WebInspect can be configured to submit data automatically for any form encountered during a crawl (for example, if a page requires entry of a telephone number, etc.).

SSL Support

Fortify WebInspect can crawl any site that uses SSL and determine whether data is being properly encrypted and protected.

Proxy Support

A proxy server can be used to ensure network security, provide adequate caching purposes, and regulate administrative control. Fortify WebInspect can crawl sites that use a proxy server, and check for vulnerabilities specifically related to that configuration.

Client Certificate Support

A certificate is a statement verifying the identity of a person or the security of a website. Attackers will attempt to alter the values of client certificates to gain unauthorized access to your web application.

State Management

State is a property of connectivity. HTTP is a stateless protocol; no concept of session state is maintained by HTTP when handling client-server communications. Fortify WebInspect determines if any cookies used on your web application are secure (are they set to expire, properly handled, etc.), and if session IDs are managed securely.

Directory Enumeration

Directory enumeration lists all directory paths and possibilities on the application server, including hidden directories that could possibly contain sensitive information. Fortify WebInspect uses a database of known folders (such as admin, test, logs, etc.) and hidden areas discovered during a crawl when composing a directory enumeration listing.

Web Server Assessment

During a web server assessment, Fortify WebInspect test your proprietary web server for vulnerabilities utilizing information gathered during a Site Search and other applied methodologies. Protocol and extension implementation analysis is used to determine what services the server offers, whether or not they conform to established standards for these services, and details regarding their implementation. As web server configurations are responsible for serving content and launching applications, damage from an attack on an unprotected proprietary web server can include denial of service, the posting of inappropriate messages or graphics on the site, deletion of files, or damaging code or software packages being left on the server.

HTTP Compliance

HTTP compliance testing assesses the web server or proxy server for proper compliance to HTTP/1.0 and HTTP/1.1 rules. This testing consists of attacks such as sending a data buffer larger than the marked length (buffer overflows). Servers are tested to see if they properly sanitize data by mixing and matching various methods and headers that are never seen within a normal request and determining if the web server handles the requests properly. These attacks can determine if a web server or web device complies with HTTP specifications and can also uncover unknown vulnerabilities.

WebDAV Compliance

WebDAV allows users to place and manipulate files in a directory on your web server. Fortify WebInspect determines whether or not WebDAV privileges can be exceeded and manipulated on your web server.

SSL Strength

SSL strength identification determines the encryption level accepted by a web server. This can be important to ensure that secure clients do not connect at an encryption level lower than the expected standard, and that data is being properly encrypted to prevent its interception.

Certificate Analysis

Fortify WebInspect analyzes the SSL certificate for improper properties such as unknown CA certificate analysis or expired time.

HTTP Method Support

Fortify WebInspect determines which HTTP methods are supported by the web server.

Example: Does the webserver support GET,PUT,INDEX,POST,CONNECT, etc.

Content Investigation

Content Investigation involves searching through content discovered during a Site Search to determine what information is available to users of your web application that should remain private. Fortify WebInspect searches for the following items when conducting Content Investigation (although by no means a comprehensive list), and will determine each item's potential level of exploitation.

Spam Gateway Detection

Spam gateways are e-mail web applications that allow the client to specify the location of the mail recipient via hidden form inputs or parameters.

Client-Side Pricing

Client-side pricing is a web application flaw that allows the client to specify item pricing via hidden form inputs or parameters.

Sensitive Developer Comments

Developer comments in HTML often reveal sensitive information about an application’s internal mechanics and configuration. For example, something as seemingly innocuous as a comment referencing the required order of fields in a table could potentially give an attacker a key piece of information needed to crack the security of your site. Fortify WebInspect lists all comments found in the site's code in the Comments area on the Information pane.

WebServer/Web Package Identification

Fortify WebInspect will identify all services and banners on the web server, and ascertain the vendors and version numbers of all available software packages used by your web application. This is accomplished through a variety of methods, listed below.

- Header Evidence - For example, Server: Microsoft-IIS/5.0
- Link Evidence - For example, <a href="/webapp.php"> indicates that the PHP web application server is running.
- Default/Template Page Evidence - For example, "If you can see this, it means that the installation of the Apache web server software on this system was successful."

Absolute Path Detection

Fortify WebInspect detects whether a fully qualified pathname was able to be discovered anywhere within an application. Certain vulnerabilities can only be exploited if the attacker has the fully qualified pathname.

Example: /opt/Web/docroot/, c:\inetpub\wwwroot"

Error Message Identification

Often, error messages will reveal more than they were designed to do. For example, pages containing /servletimages/logo2circle.gif are default template BEA WebLogic error pages. An attacker forearmed with that knowledge can customize his attack to take advantage of that server's inherent vulnerabilities.

Permissions Assessment

Fortify WebInspect will determine what level of permissions (such as uploading files to the web server, editing data, traversing directories, etc.) are available in different areas of your web application, and then determine the best way to remedy any inherent security vulnerabilities.

Brute Force Authentication Attacks

Brute force attacks test for susceptibility to dictionary attacks (files containing common logons and passwords). Fortify WebInspect tests Basic, NTLM, and web form authentication for susceptibility to a brute force attack.

Known Attacks

Known attacks include all exploitable holes and bugs in web servers, applications, and other third-party components that have been published, posted, or otherwise communicated. Most of these vulnerabilities have existing patches, but hackers will exploit systems where patches have not been installed in a timely fashion. Known attack information is included in all other methodologies.

Fortify WebInspect rely on a proprietary database that contains fingerprints of known attacks dating back to the birth of the World Wide Web. They check for and download new risks and exploits each they run, ensuring that the product is always updated and at the forefront of hacking expertise.

Policies

Each policy is kept up to date through the Smart Update function, ensuring that scans are accurate and capable of detecting the most recently discovered threats. Fortify WebInspect contains the following packaged policies that you can use with your scans and crawls to determine the vulnerability of your web application.

Best Practices

The Best Practices group contains policies designed to test applications for the most pervasive and problematic web application security vulnerabilities.

- **General Data Protection Regulation (GDPR)**: The EU General Data Protection Regulation (GDPR) replaces the Data Protection Directive 95/46/EC and provides a framework for organizations on how to handle personal data. The GDPR articles that pertain to application security and require businesses to protect personal data during design and development of their products and services are as follows:

  - Article 25, data protection by design and by default, which requires businesses to implement appropriate technical and organizational measures for ensuring that, by default, only personal data that is necessary for each specific purpose of the processing is processed.

  - Article 32, security of processing, which requires businesses to protect their systems and applications from accidental or unlawful destruction, loss, alteration, unauthorized disclosure of or access to personal data.

  This policy contains a selection of checks to help identify and protect personal data specifically related to application security for the GDPR.
- **OWASP Top 10 *<year>***: This policy provides a minimum standard for web application security. The OWASP Top 10 represents a broad consensus about the most critical web application security flaws. Adopting the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces secure code. Multiple releases of the OWASP Top Ten policy may be available. For more information, consult the OWASP Top Ten Project.
- **SANS Top 25**: The SANS Top 25 Most Dangerous Software Errors provides an enumeration of the most widespread and critical errors, categorized by Common Weakness Enumeration (CWE) identifiers, that lead to serious vulnerabilities in software. These software errors are often easy to find and exploit. The inherent danger in these errors is that they can allow an attacker to take over the software completely, steal data, or prevent the software from working altogether.
- **Standard**: A standard scan includes an automated crawl of the server and performs checks for known and unknown vulnerabilities such as SQL Injection and Cross-Site Scripting as well as poor

error handling and weak SSL configuration at the web server, web application server, and web application layers.

By Type

The By Type group contains policies designed with a specific application layer, type of vulnerability, or generic function as its focus. For instance, the Application policy contains all checks designed to test an application, as opposed to the operating system.

- **Aggressive SQL Injection**: This policy performs a comprehensive security assessment of your web application for SQL Injection vulnerabilities. SQL Injection is an attack technique that takes advantage of non-validated input vulnerabilities to pass arbitrary SQL queries and/or commands through the web application for execution by a backend database. This policy performs a more accurate and decisive job, but has a longer scan time.
- **Apache Struts**: This policy detects supported known advisories against the Apache Struts framework.
- **Blank**: This policy is a template that you can use to build your own policy. It includes an automated crawl of the server and no vulnerability checks. Edit this policy to create custom policies that only scan for specific vulnerabilities.
- **Client-side**: This policy intends to detect all issues that require an attacker to perform phishing in order to deliver an attack. These issues are typically manifested on the client, thus enforcing the phishing requirement. This includes Reflected Cross-site Scripting and various HTML5 checks. This policy may be used in conjunction with the Server-side policy to provide coverage across both the client and the server.
- **Criticals and Highs**: Use the Criticals and Highs policy to quickly scan your web applications for the most urgent and pressing vulnerabilities while not endangering production servers. This policy checks for SQL Injection, Cross-Site Scripting, and other critical and high severity vulnerabilities. It does not contain checks that may write data to databases or create denial-of-service conditions, and is safe to run against production servers.
- **Cross-Site Scripting**: This policy performs a security scan of your web application for cross-site scripting (XSS) vulnerabilities. XSS is an attack technique that forces a website to echo attacker-supplied executable code, such as HTML code or client-side script, which then loads in a user's browser. Such an attack can be used to bypass access controls or conduct phishing expeditions.
- **DISA STIG *<version>***: The Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) provides security guidance for use throughout the application development lifecycle. This policy contains a selection of checks to help the application meet the secure coding requirements of the DISA STIG *<version>*. Multiple versions of the DISA STIG policy may be available.
- **Mobile**: A mobile scan detects security flaws based on the communication observed between a mobile application and the supporting backend services.
- **NoSQL and Node.js**: This policy includes an automated crawl of the server and performs checks for known and unknown vulnerabilities targeting databases based on NoSQL, such as MongoDB, and server side infrastructures based on JavaScript, such as Node.js.
- **Passive Scan**: The Passive Scan policy scans an application for vulnerabilities detectable without active exploitation, making it safe to run against production servers. Vulnerabilities detected by this

policy include issues of path disclosure, error messages, and others of a similar nature.

- **Privilege Escalation**: The Privilege Escalation policy scans your web application for programming errors or design flaws that allow an attacker to gain elevated access to data and applications. The policy uses checks that compare responses of identical requests with different privilege levels.
- **Server-side**: This policy contains checks that target various issues on the server-side of an application. This includes various injection attacks, transport layer security, and privacy violation, but does not include attack surface discovery such as directory enumeration or backup file search. All vulnerabilities detected by this policy may be directly targeted by an attacker. This policy may be used in conjunction with the Client-side policy to provide coverage across both the client and the server.
- **SQL Injection**: The SQL Injection policy performs a security scan of your web application for SQL injection vulnerabilities. SQL injection is an attack technique that takes advantage of non-validated input vulnerabilities to pass arbitrary SQL queries and/or commands through the web application for execution by a backend database.
- **Transport Layer Security**: This policy performs a security assessment of your web application for insecure SSL/TLS configurations and critical transport layer security vulnerabilities, such as Heartbleed, Poodle, and SSL Renegotiation attacks.
- **WebSocket**: This policy detects vulnerabilities related to WebSocket implementation in your application.

Custom

The Custom group contains all user-created policies and any custom policies modified by a user.

Hazardous

The Hazardous group contains a policy with potentially dangerous checks, such as a denial-of-service attack, that could cause production servers to fail. Use this policy against non-production servers and systems only.

- **All Checks**: An All Checks scan includes an automated crawl of the server and performs all active checks from SecureBase, the database. This scan includes all checks that are listed in the compliance reports that are available in Fortify web application and web services vulnerability scan products. This includes checks for known and unknown vulnerabilities at the web server, web application server, and web application layers.

  > **Caution!** An All Checks scan includes checks that may write data to databases, submit forms, and create denial-of-service conditions. Fortify strongly recommends using the All Checks policy only in test environments.

Deprecated Checks and Policies

The following policies and checks are deprecated and are no longer maintained.

- **Application (Deprecated)**: The Application policy performs a security scan of your web application by submitting known and unknown web application attacks, and only submits specific attacks that assess the application layer. When performing scans of enterprise level web applications, use the Application Only policy in conjunction with the Platform Only policy to optimize your scan in terms of speed and memory usage.

- **Assault (Deprecated)**: An assault scan includes an automated crawl of the server and performs checks for known and unknown vulnerabilities at the web server, web application server, and web application layers. An assault scan includes checks that can create denial-of-service conditions. It is strongly recommended that assault scans only be used in test environments.
- **Deprecated Checks**: As technologies go end of life and fade out of the technical landscape it is necessary to prune the policy from time to time to remove checks that are no longer technically necessary. Deprecated checks policy includes checks that are either deemed end of life based on current technological landscape or have been re-implemented using smart and efficient audit algorithms that leverage latest enhancements of core WebInspect framework.
- **Dev (Deprecated)**: A Developer scan includes an automated crawl of the server and performs checks for known and unknown vulnerabilities at the web application layer only. The policy does not execute checks that are likely to create denial-of-service conditions, so it is safe to run on production systems.
- **OpenSSL Heartbleed (Deprecated)**: This policy performs a security assessment of your web application for the critical TLS Heartbeat read overrun vulnerability. This vulnerability could potentially disclose critical server and web application data residing in the server memory at the time a malicious user sends a malformed Heartbeat request to the server hosting the site.
- **OWASP Top 10 Application Security Risks - 2010 (Deprecated)**: This policy provides a minimum standard for web application security. The OWASP Top 10 represents a broad consensus about what the most critical web application security flaws are. Adopting the OWASP Top 10 is perhaps the most effective first step towards changing the software development culture within your organization into one that produces secure code. This policy includes elements specific to the 2010 Top Ten list. For more information, consult the OWASP Top Ten Project.
- **Platform (Deprecated)**: The Platform policy performs a security scan of your web application platform by submitting attacks specifically against the web server and known web applications. When performing scans of enterprise-level web applications, use the Platform Only policy in conjunction with the Application Only policy to optimize your scan in terms of speed and memory usage.
- **QA (Deprecated)**: The QA policy is designed to help QA professionals make project release decisions in terms of web application security. It performs checks for both known and unknown web application vulnerabilities. However, it does not submit potentially hazardous checks, making it safe to run on production systems.
- **Quick (Deprecated)**: A Quick scan includes an automated crawl of the server and performs checks for known vulnerabilities in major packages and unknown vulnerabilities at the web server, web application server and web application layers. A quick scan does not run checks that are likely to create denial-of-service conditions, so it is safe to run on production systems.
- **Safe (Deprecated)**: A Safe scan includes an automated crawl of the server and performs checks for most known vulnerabilities in major packages and some unknown vulnerabilities at the web server, web application server and web application layers. A safe scan does not run any checks that could potentially trigger a denial-of-service condition, even on sensitive systems.
- **Standard (Deprecated)**: Standard (Deprecated) policy is copy of the original standard policy before it was revamped in R1 2015 release. A standard scan includes an automated crawl of the server and performs checks for known and unknown vulnerabilities at the web server, web application server and web application layers. A standard scan does not run checks that are likely to create denial-of-

service conditions, so it is safe to run on production systems.

See Also

"About the Policy Manager Tool" on page 56

Policy Manager Icons

The following table describes the icons that are used in the Policy Manager tree view.

Icon	Definition
🌐	The policy.
🗂	Attack Group Folder: Folders that contain vulnerability assessments.
🧩	Audit Methodology: A set of checks that compose an audit methodology. For example, Site Search is part of the Audit methodology. For more information on methodologies, see "Methodologies" on page 70.
❗	A critical vulnerability wherein an attacker might have the ability to execute commands on the server or retrieve and modify private information.
🔴	A high vulnerability. Generally, the ability to view source code, files out of the web root, and sensitive error messages.
🟡	A medium vulnerability. Indicates non-HTML errors or issues that could be sensitive.
🔵	A low vulnerability. Indicates interesting issues or issues that could potentially become higher ones.

Audit Engines

Fortify WebInspect uses the following audit engines:

- **Adaptive Agents**: Certain vulnerabilities require a large amount of logic when checking for them. For example, a buffer overflow JRun check might cause a server to crash if conducted through a vulnerability database. Instead, an adaptive agent with the proper amount of logic can be written to prevent such a problem. With this smart approach, Fortify WebInspect continuously applies appropriate assessment resources that adapt to the specific application environment.
- **Arbitrary Remote File Include**: This engine checks for vulnerabilities that may allow fetching and incorporating data from arbitrary URLs supplied by an attacker.
- **Comment Checks**: The comment audit examines each session for file names and/or URLs in

comments. Upon finding a filename or URL, the audit will check to see if the file or URL exists.

- **Cookie Injection**: Cookies and headers are just as vulnerable to injection attacks as text fields in forms. Cookie injection occurs when unvalidated data is sent by a user's browser as part of a cookie. The Cookie Injection audit engine attempts certain traditional parameter injection attacks against different cookie values
- **Cross-Site Scripting**: This engine conducts cross-site scripting parameter injection attacks. Applications are vulnerable to these attacks when developers do not adequately filter or verify client-supplied data that is returned by the application to the server.
- **Directory Enumeration**: Directory Enumeration finds all directory paths and possibilities on the application server, including hidden directories which could possibly contain sensitive information. This helps Fortify WebInspect create a full and accurate map of the targeted site.
- **Directory Extension Addition**: Directory extension checking involves adding extensions to directories and removing the trailing slash to find archived directories left on the server. Fortify WebInspect attempts to locate all directories that have been left on your server that could be used by an attacker.
- **File Extension**: Network administrators and developers often leave backup files and scripts on the web server. These files commonly contain information that can be used to breach a site's security. Extension checking involves replacing extensions on files, and then looking for older or backup versions stored on the site. For example, an attacker who finds hi.asp might search for hi.old and hi.back, and retrieve the script's source code. Fortify WebInspect attempts to locate all files that could be utilized by an attacker that have been left on your server.
- **File Prefix**: Network administrators and developers often leave backup files and scripts on the web server. These files commonly contain information that can be used to breach a site's security. Prefix checking involves affixing a value to file names, and then looking for older or backup versions stored on the site.
- **File Suffix**: File suffix checking involves affixing a value to file names, and then looking for older or backup versions stored on the site. See File Prefix above.
- **Fixed Checks**: This audit performs checks for files with known vulnerabilities. This audit is the same as the ABS Checks audit, with the exception being that the Fixed Checks audit does not probe the directory structure before sending the attacks.
- **FlashStaticAnalysis**: Performs Flash source code analysis to detect vulnerabilities.
- **Fortify Agent Probe Engine**: This engine sends probes for hints whether a particular parameter or injection point would be vulnerable to the attack suggestions provided in the audit inputs.
- **Header Injection**: Cookies and headers are just as vulnerable to injection attacks as text fields in forms. HTTP header injection occurs when HTTP headers are dynamically generated with user input that includes malicious content. The Header Injection audit engine attempts certain traditional parameter injection attacks against different types of HTTP headers.
- **Keyword Search**: Information disclosure attacks focus on ways of getting a website to reveal system-specific information or confidential data, including user data, that should not be exposed to anonymous users. The Keyword Search audit engine examines every response from the web server for information, such as error messages, directory listings, credit card numbers, etc., that is not properly protected by the website
- **Known Vulnerabilities**: This engine checks for files with known vulnerabilities. The audit will perform a probe of directories known to contain these files and then send requests based on any

discovered directories.

- **Local File Inclusion**: Local file reading/inclusion vulnerabilities exist when an attacker can influence the application to read (presumably arbitrary) files specified by the attacker. The engine submits to the web application various values that contain various combinations of relative and absolute file names for specific known files. The engine considers the attack a success if the contents of those files are displayed.
- **Persistent Cross-Site Scripting**: This engine must be enabled to check for Persistent Cross-Site Scripting vulnerabilities (also known as Stored Cross-Site Scripting). When successfully exploited, Persistent Cross-Site Scripting can allow an attacker to inject malicious scripts into the target application's client-side code.
- **Postdata Injection**: Since manipulating a query string is as easy as typing text in the address bar of a browser, many web applications rely on the POST method coupled with the use of forms (rather than GET) to pass data between pages. Since browsers normally don’t display POST data, some programmers are lulled into thinking that it is difficult or impossible to change the data, when in fact the opposite is true. Fortify WebInspect determines your application’s susceptibility to attacks that rely on the POST method of parameter manipulation.
- **Postdata Sequence**: Since manipulating a query string is as easy as typing text in the address bar of a browser, many web applications rely on the POST method coupled with the use of forms (rather than GET) to pass data between pages. Since browsers normally don’t display POST data, some programmers are lulled into thinking that it is difficult or impossible to change the data, when in fact the opposite is true. Fortify WebInspect determines your application’s susceptibility to attacks that rely on the POST method of parameter manipulation by sending fragmented data to the target.
- **Query Injection**: Web applications often use query strings as a simple method of passing data from the client to the server. Query strings are a way to add data calls to a hyperlink, and then retrieve that information on the linked page when it is displayed. By manipulating query strings, an attacker can easily steal information from a database, learn details about the architecture of your web application, or possibly execute commands on your web server.

  When conducting an audit, Fortify WebInspect implements advanced query string manipulation to ascertain the feasibility of command execution on your server(s), and determines the vulnerability of your web applications to query string manipulation.
- **Query Sequence**: Web applications often use query strings as a simple method of passing data from the client to the server. Query strings are a way to add data calls to a hyperlink, and then retrieve that information on the linked page when it is displayed. By manipulating query strings, an attacker can easily steal information from a database, learn details about the architecture of your web application, or possibly execute commands on your web server.

  When conducting an audit, Fortify WebInspect implements advanced query string manipulation to ascertain the feasibility of command execution on your server(s), and determines the vulnerability of your web applications to query string manipulation by sending fragmented data to the target.
- **Reclassify**: This engine analyzes the responses to generic/application non-specific attacks and reclassifies certain vulnerability instances into specific known application vulnerabilities.
- **Request Modification**: Several types of attacks involve malformed requests that result in a failed response from the web server. The Request Modification engine generates requests that are derived from other requests that match a pattern, and then evaluates the response to determine if these types of attacks are possible.
- **Site Search**: This can be considered the information gathering stage, much as an attacker would

learn as much as possible about your web application before launching an attack. Site search is used to locate resources such as documents, applications and directories on the server that are not intended to be viewed by web users. Disclosure of such resources can result in the disclosure of confidential data, information about internal server and application configurations and settings, administrative access to the site, and information about application source code.

- **SOAP Assessment**: Web services are programs that communicate with other applications (rather than with users) and answer requests for information. Most web services utilize SOAP (Simple Object Access Protocol) to send XML data between the web service and the client web application making the information request. SOAP assessment involves checking for security vulnerabilities inherent within that transport mechanism.
- **SQL Injection**: SQL Injection is an attack in which hackers use SQL statements via an Internet browser to extract, add, or modify data, create a denial of service, bypass authentication, or execute remote commands. The SQL Injection engine detects the following attacks:
  - Injection through user input, such as malicious strings in web forms
  - Injection through cookies, such as modified cookie fields that contain attack strings
  - Injection through server variables, such as headers that are manipulated to contain attack strings

Audit Options

Fortify WebInspect uses the following audit options.

- **CVS Entries Parser**: This engine parses any Entries files found within the scan for links to add to the crawler engine.
- **Robots.txt Parser**: This engine parses any robots.txt files found within the scan for links to add to the crawler engine.
- **WebInspect Scan Signature**: This signature sends the text SCANNED-BY-HP- to the server. The text appears in the webserver logs and indicates that a scan has occurred.
- **Ws_ftp.log Parser**: This engine parses any Ws_ftp.log files it finds and will add links to the site directory tree.

General Application Testing

This group of checks is applicable to all web applications generally. It includes Directory Enumeration, which looks for common directories in the root of the server. It also includes input injection checks such as SQL Injection and Cross-Site Scripting.

Third-Party Web Applications

This group of checks looks for known vulnerabilities associated with third-party web applications.

Web Frameworks/Languages

This group of agents looks for known vulnerabilities associated with web application servers. It also determines if known flaws in certain scripting languages can be exploited on the target system.

Web Servers

This group of agents looks for known vulnerabilities associated with the following web servers:

- Apache
- IIS
- Lotus Domino
- Minor (a collection of servers including ATPhttpd, 4D, Abyss, Alibaba, BadBlue, and others)
- Netscape/iPlanet
- Secure IIS
- Website Pro
- WebSphere Proxy
- Zeus

For detailed information about all the possible agents, expand the Web Servers node and click on any agent.

Custom Agents

Even though Fortify WebInspect launches thousands of agents to assess your web application during a normal scan, a developer may want to check for a specific condition that is unique to your environment or application. The developer may create a custom agent using the WebInspect Software Developer's Kit (SDK). You may then integrate the custom agent into one or more policies using the Policy Manager.

See Also

"Using a Custom Agent" on page 70

Custom Checks

A custom check is a user-defined probe for a specific vulnerability that the standard repertoire does not address. A custom check can be created using a simple wizard.

See Also

"Creating a Custom Check" on page 60

Regular Expressions

Special characters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used. Another recommended resource is the Regular Expression Library.

Character	Description
\	Marks the next character as special. /n/ matches the character " n ". The sequence /\n/ matches a linefeed or newline character.
^	Matches the beginning of input or line. Also used with character classes as a negation character. For example, to exclude everything in the content directory except /content/en and /content/ca, use: /content/[^(en|ca)].*/.* . Also see \S \D \W.
$	Matches the end of input or line.
*	Matches the preceding character zero or more times. /zo*/ matches either " z " or "zoo."
+	Matches the preceding character one or more times. /zo+/ matches "zoo" but not "z."
?	Matches the preceding character zero or one time. /a?ve?/ matches the "ve" in "never."
.	Matches any single character except a newline character.
[xyz]	A character set. Matches any one of the enclosed characters. /[abc]/ matches the "a" in "plain."
\b	Matches a word boundary, such as a space. /ea*r\b/ matches the "er" in "never early."
\B	Matches a non-word boundary. /ea*r\B/ matches the "ear" in "never early."
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a non-digit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a linefeed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [

Character	Description
	\f\n\r\t\v]
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v]
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any non-word character. Equivalent to [^A-Za-z0-9_].

Regular Expression Extensions

Micro Focus engineers have developed and implemented extensions to the normal regular expression syntax. When building a regular expression, you can use the following tags and operators.

Regular Expression Tags

- [ALL]
- [BODY]
- [STATUSLINE]
- [HEADERS]
- [STATUSCODE]
- [STATUSDESCRIPTION]
- [COOKIES]

Regular Expression Operators

- AND
- OR
- NOT
- [ ]
- ( )

Examples

- To detect a response in which (a) the status line contains a status code of "200" and (b) the phrase "logged out" appears anywhere in the message body, use the following regular expression:

  [STATUSCODE]200 AND [BODY]logged\sout

- To detect a response indicating that the requested resource resides temporarily under a different URI (redirection) and has a reference to the path "/Login.asp" anywhere in the response, use the following:

  [STATUSCODE]302 AND [ALL]Login.asp

- To detect a response containing either (a) a status code of "200" and the phrase "logged out" or "session expired" anywhere in the body, or (b) a status code of "302" and a reference to the path "/Login.asp" anywhere in the response, use the following regular expression:

  ([STATUSCODE]200 AND [BODY]logged\sout OR [BODY]session\sexpired) OR ([STATUSCODE]302 AND [ALL]Login.asp)

  Note: You must include a space (ASCII 32) before and after an "open" or "close" parenthesis; otherwise, the parenthesis will be erroneously considered as part of the regular expression.

- To detect a redirection response where "login.aspx" appears anywhere in the redirection Location header, use the following regular expression:

  [STATUSCODE]302 AND [HEADERS]Location:\slogin.aspx

- To detect a response containing a specific string (such as "Please Authenticate") in the Reason-Phrase portion of the status line, use the following regular expression:

  [STATUSDESCRIPTION]Please\sAuthenticate

Chapter 8: About the Regular Expression Editor Tool

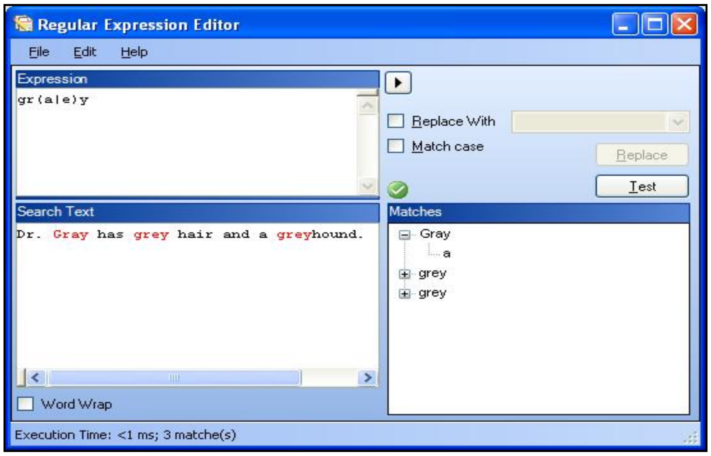
A regular expression is a pattern that describes a set of strings. Regular expressions are constructed similarly to mathematical expressions by using various operators to combine smaller expressions. Only advanced users with a working knowledge of regular expressions should use this feature.

Testing a Regular Expression

Use the Regular Expression Editor to test and verify regular expressions, as follows:

1. Click **Tools > Regular Expression Editor**.

   The Regular Expression Editor window opens.



Expression

```
gr(a|e)y
```

□ Replace With

□ Match case

Search Text

```
Dr. Gray has grey hair and a greyhound.
```

Matches

```
⊟ Gray
    a
⊞ grey
⊞ grey
```

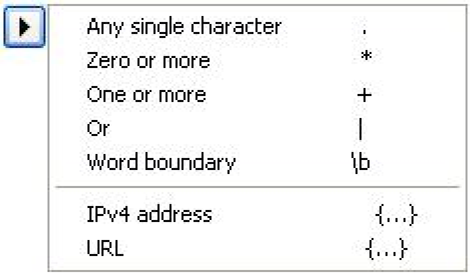
□ Word Wrap

Execution Time: <1 ms; 3 matche(s)

2. In the **Expression** area, type or paste a regular expression that you think will find the text for which you are searching.

   For assistance, click ▶ to reveal a list of objects. These include metacharacters and regular expressions that define a URL and an IP address. Click an object to insert it.

   > **Note:** You can also use Regular Expression Extensions to restrict your search to certain areas of an HTTP message.



The Regular Expression Editor examines the syntax of the entered expression and displays ✓ (if valid) or ✗ (if invalid).

3. In the **Search Text** area, type (or paste) the text through which you want to search.

   Alternatively, you can load an HTTP request or response message that you previously saved using the HTTP Editor, as follows:

   a. Click **File > Open Request**.

      The Request file is actually a session containing data for both the HTTP request and response.

   b. Using the standard file-selection window, choose a file containing the saved session.

   c. Select either **Request** or **Response**.

   d. Click **OK**.

4. To find only those occurrences matching the case of the expression, select the **Match Case** check box.

5. To substitute the string identified by the regular expression with a different string:

   a. Select the **Replace With** check box.

   b. Type or select a string using the drop-down combo box.

6. Click **Test** to search the target text for strings that match the regular expression. Matches are highlighted in red.

7. If you selected the **Replace** option, click **Replace** to substitute all found strings with the replacement string.

See Also

"Regular Expressions" below

"Regular Expression Extensions" on page 94

Regular Expressions

Special characters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used. Another recommended resource is the Regular Expression Library.

Character	Description
\	Marks the next character as special. /n/ matches the character " n ". The sequence /\n/

Character	Description
	matches a linefeed or newline character.
^	Matches the beginning of input or line. Also used with character classes as a negation character. For example, to exclude everything in the content directory except /content/en and /content/ca, use: /content/[^(en|ca)].*/.* . Also see \S \D \W.
$	Matches the end of input or line.
*	Matches the preceding character zero or more times. /zo*/ matches either " z " or "zoo."
+	Matches the preceding character one or more times. /zo+/ matches "zoo" but not "z."
?	Matches the preceding character zero or one time. /a?ve?/ matches the "ve" in "never."
.	Matches any single character except a newline character.
[xyz]	A character set. Matches any one of the enclosed characters. /[abc]/ matches the "a" in "plain."
\b	Matches a word boundary, such as a space. /ea*r\b/ matches the "er" in "never early."
\B	Matches a non-word boundary. /ea*r\B/ matches the "ear" in "never early."
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a non-digit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a linefeed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v]
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v]
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any non-word character. Equivalent to [^A-Za-z0-9_].

Regular Expression Extensions

Micro Focus engineers have developed and implemented extensions to the normal regular expression syntax. When building a regular expression, you can use the following tags and operators.

Regular Expression Tags

- [BODY]
- [STATUSCODE]
- [STATUSDESCRIPTION]
- [STATUSLINE]
- [HEADERS]
- [ALL]
- [COOKIES]
- [SETCOOKIES]
- [METHOD]
- [REQUESTLINE]
- [VERSION]
- [POSTDATA]
- [URI]
- [TEXT]

Regular Expression Operators

- AND
- OR
- NOT
- [ ]
- ( )

Examples

- To detect a response in which (a) the status line contains a status code of "200" and (b) the phrase "logged out" appears anywhere in the message body, use the following regular expression:

  [STATUSCODE]200 AND [BODY]logged\sout
- To detect a response indicating that the requested resource resides temporarily under a different URI (redirection) and has a reference to the path "/Login.asp" anywhere in the response, use the following:

[STATUSCODE]302 AND [ALL]Login.asp

- To detect a response containing either (a) a status code of "200" and the phrase "logged out" or "session expired" anywhere in the body, or (b) a status code of "302" and a reference to the path "/Login.asp" anywhere in the response, use the following regular expression:

  ([STATUSCODE]200 AND [BODY]logged\sout OR [BODY]session\sexpired) OR ([STATUSCODE]302 AND [ALL]Login.asp)

  > **Note:** You must include a space (ASCII 32) before and after an "open" or "close" parenthesis; otherwise, the parenthesis will be erroneously considered as part of the regular expression.

- To detect a redirection response where "login.aspx" appears anywhere in the redirection Location header, use the following regular expression:

  [STATUSCODE]302 AND [HEADERS]Location:\slogin.aspx

- To detect a response containing a specific string (such as "Please Authenticate") in the Reason-Phrase portion of the status line, use the following regular expression:

  [STATUSDESCRIPTION]Please\sAuthenticate

Chapter 9: About the Server Analyzer Tool (Fortify WebInspect Only)

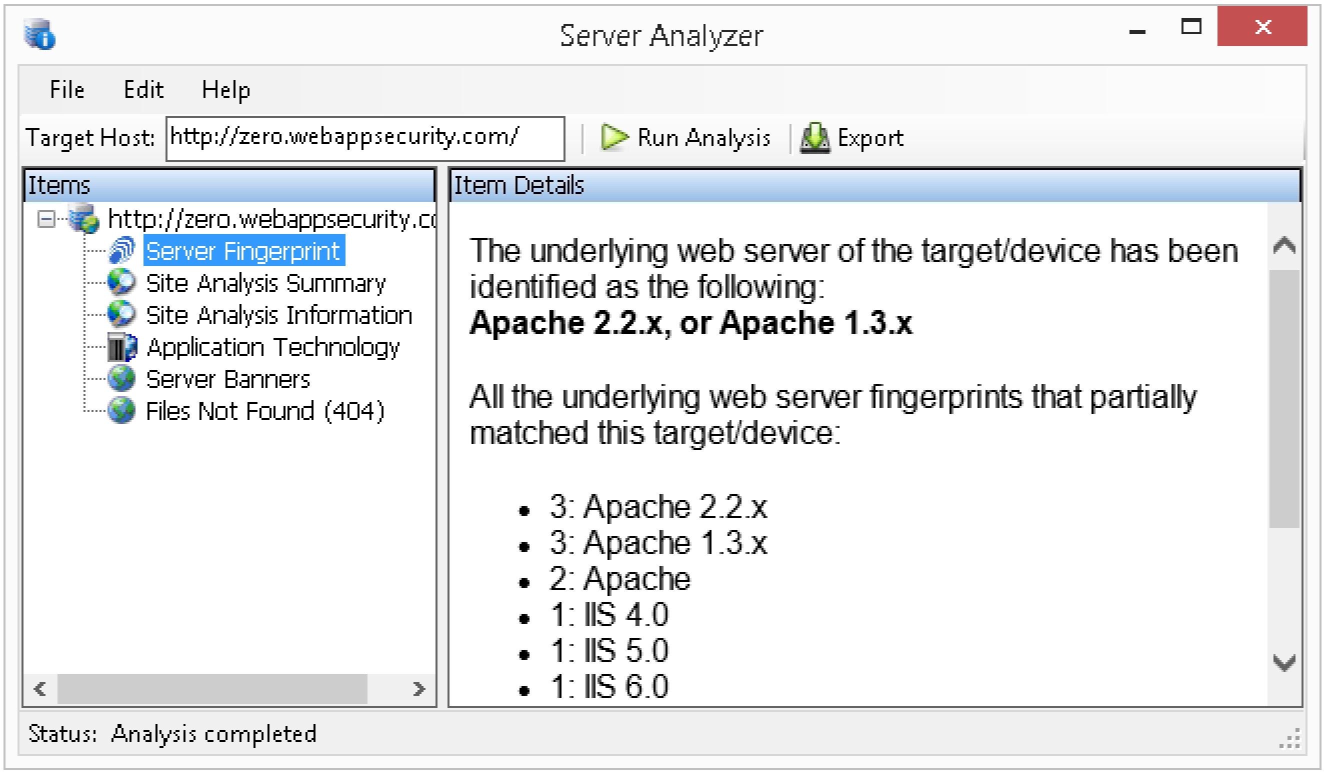
The Server Analyzer interrogates a server to reveal the server's operating system, banners, cookies, and other information.

Analyzing a Server

To analyze a server:

1. In the **Target Host** field, enter the URL or IP address of the target server.
2. If host authentication (user name and password) is required, or if you are accessing the target server through a proxy server, click **Edit > Settings**, and enter the requested information. For more information, see "Authentication Settings" on the next page and "Proxy Settings" on page 98.
3. Click the **Run Analysis** icon.

   When finished, the Server Analyzer displays the status "Analysis completed" and a list of items that were analyzed.
4. Select an item in the **Item** pane to view its information in the **Item Details** pane.



Target Host: http://zero.webappsecurity.com/

http://zero.webappsecurity.co

The underlying web server of the target/device has been identified as the following:
Apache 2.2.x, or Apache 1.3.x

All the underlying web server fingerprints that partially matched this target/device:

- 3: Apache 2.2.x
- 3: Apache 1.3.x
- 2: Apache
- 1: IIS 4.0
- 1: IIS 5.0
- 1: IIS 6.0

Status: Analysis completed

Modifying Settings

To modify the Server Analyzer settings:

1. Click **Edit > Settings**.
2. Select one of the following:
   - **Host Authentication**. See "Authentication Settings" below.
   - **Proxy**. See "Proxy Settings" on the next page.
3. Click **OK**.

Exporting Analyzer Results

To export the results of the analysis to an HTML file:

1. Click **File > Export**.
2. On the Export File window, select or enter a location and file name.
3. Click **Save**.

See Also

"Authentication Settings" below

"Proxy Settings" on the next page

Authentication Settings

Authentication Method

If authentication is required, select the authentication type:

- **Automatic** - If you select this option, the Server Analyzer tool will determine the correct authentication method.

  > **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

- **HTTP Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.

- **NTLM** - NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

  Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and Server Analyzer has to pass through a proxy server to submit its requests to the web server, Server Analyzer may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.
- **Digest** - The Windows Server operating system implements the Digest authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using Digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.
- **Kerberos** - Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

Authentication Credentials

Type a user ID in the **User name** field and the user's password in the **Password** field. To prevent mistyping, repeat the password in the **Confirm Password** field.

To use these credentials whenever the Server Analyzer encounters a password input control, select **Submit these credentials to forms with password input fields**.

Proxy Settings

To access this feature, click **Edit > Settings**. Then select **Proxy**.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

If you select this option, Server Analyzer will use the Web Proxy Autodiscovery Protocol (WPAD) to locate and use a proxy autoconfig file to configure the browser's web proxy settings.

Use System Proxy Settings

Select this option to import your proxy server information from the local machine.

Use Firefox proxy settings

Select this option to import your proxy server information from Firefox.

> **Note:** Using browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.

Configure proxy using a PAC file

Select this option to load proxy settings from a Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** field.

Explicitly configure proxy

Select this option to access the Internet through a proxy server, and then enter the requested information:

1. In the **Server** field, type the URL or IP address of your proxy server, followed (in the **Port** field) by the port number (for example, 8080).
2. Select a protocol **Type** for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or Standard.
3. If authentication is required, select a type from the **Authentication** list:
    - **Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

      The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools.

Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.

- **NTLM** - NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

  Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and Server Analyzer has to pass through a proxy server to submit its requests to the web server, Server Analyzer may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

- **Kerberos** - Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

- **Digest** - The Windows Server operating system implements the Digest authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using Digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

- **Automatic** - Allow Server Analyzer to determine the correct authentication type.

  Note: Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

- **Negotiate** - The Negotiate authentication protocol begins with the option to negotiate for an authentication protocol. When the client requests access to a service, the server replies with a list of authentication protocols that it can support and an authentication challenge based on the protocol that is its first choice. For example, the server might list Kerberos and NTLM, and send a Kerberos challenge. The client examines the contents of the reply and checks to determine whether it supports any of the specified protocols. If the client supports the preferred protocol, authentication proceeds. If the client does not support the preferred protocol, but does support one of the other protocols listed by the server, the client lets the server know which authentication protocol it supports, and the authentication proceeds. If the client does not support any of the listed protocols, the authentication exchange fails.

4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** field. Use commas to separate entries.

Specify Alternative Proxy for HTTPS

For proxy servers accepting HTTPS connections, select the **Specify Alternative Proxy for HTTPS** check box and provide the requested information.

Chapter 10: Server Profiler

Use the Server Profiler to conduct a preliminary examination of a Web site to determine if certain Fortify WebInspect settings should be modified. If changes appear to be required, the Profiler returns a list of suggestions, which you may accept or reject.

For example, the Server Profiler may detect that authorization is required to enter the site, but you have not specified a valid user name and password. Rather than proceed with a scan that would return significantly diminished results, you could follow the Server Profiler's prompt to configure the required information before continuing.

Similarly, your settings may specify that Fortify WebInspect should not conduct "file-not-found" detection. This process is useful for Web sites that do not return a status "404 Not Found" when a client requests a resource that does not exist (they may instead return a status "200 OK," but the response contains a message that the file cannot be found). If the Profiler determines that such a scheme has been implemented in the target site, it would suggest that you modify the Fortify WebInspect setting to accommodate this feature.

The Server Profiler can be selected during a Guided Scan, or enabled in the Application settings.

Using the Server Profiler

You can use either of two methods to invoke the Server Profiler:

Launch Server Profiler as a Tool

Follow these steps to launch the Server Profiler:

1. Click the Fortify WebInspect **Tools** menu and select **ServerProfiler**.
2. In the **URL** box, enter or select a URL or IP address.
3. (Optional) If necessary, modify the **Sample Size**. Large Web sites may require more than the default number of sessions to sufficiently analyze the requirements.
4. Click **Analyze**.

   The Profiler returns a list of suggestions (or a statement that no modifications are necessary).
5. To reject a suggestion, clear its associated check box.
6. For suggestions that require user input, provide the requested information.
7. (Optional) To save the modified settings to a file:
   a. Click **Save Settings**.
   b. Using a standard file-selection window, save the settings to a file in your Settings directory.

Invoke Server Profiler when Starting a Scan

Follow these steps to launch the profiler when beginning a scan:

1. Start a scan using one of the following methods:
   - On the Fortify WebInspect **Start Page**, click **Start a Basic Scan**.
   - Click **File > New > Basic Scan**.
   - Click the drop-down arrow on the **New** icon (on the toolbar) and select **Basic Scan**.
   - On the Fortify WebInspect **Start Page**, click **Manage Scheduled Scans**, click **Add**, and then select **Basic Scan**.
2. On step 4 of the Scan Wizard (Detailed Scan Configuration), click **Profile** (unless **Run Profiler Automatically** is selected).

   The Profiler returns a list of suggestions (or a statement that no modifications are necessary).
3. To reject a suggestion, clear its associated check box.
4. For suggestions that require user input, provide the requested information.
5. Click **Next**.

Chapter 11: SmartUpdate

For installations connected to the Internet, the SmartUpdate feature contacts the Micro Focus data center to check for new or updated adaptive agents, vulnerability checks, and policy information. SmartUpdate will also ensure that you are using the latest version of Fortify WebInspect, and will prompt you if a newer version of the product is available for download.

You can configure Fortify WebInspect settings to conduct a SmartUpdate each time you start the application (select **Application Settings** from the **Edit** menu and choose **Smart Update**).

You can also run SmartUpdate on demand through the Fortify WebInspect user interface by selecting **Start SmartUpdate** from the Fortify WebInspect **Start Page**, by selecting **SmartUpdate** from the Tools menu, or by clicking the **SmartUpdate** button on the standard toolbar.

For installations lacking an Internet connection, see "Performing a SmartUpdate (Offline)" on the next page.

> **Caution!** For enterprise installations, if SmartUpdate changes or replaces certain files used by Fortify WebInspect, the sensor service might stop and the sensor will display a status of "off line." You must launch the Fortify WebInspect application and restart the service. To do so:
>
> 1. Click **Edit > Application Settings**.
> 2. Select **Run as a Sensor**.
> 3. Click the **Start** button in the Sensor Status area.

Performing a SmartUpdate (Internet Connected)

To perform a SmartUpdate when WebInspect is connected to the Internet:

1. Do one of the following:
   - From the toolbar, click **SmartUpdate**.
   - Select **SmartUpdate** from the **Tools** menu.
   - Select **Start SmartUpdate** from the Fortify WebInspect **Start Page**.

   If updates are available, the SmartUpdater window opens with the Summary tab in view. The Summary tab displays up to three separate collapsible panes for downloading the following:

   - New and updated checks
   - Fortify WebInspect software
   - SmartUpdate software
2. Select the check box associated with one or more of the download options.

3. (Optional) To view details about the checks being updated:
    a. Click the **Check Detail** tab.

       In the left pane is a list showing the ID, Name, and Version of checks being updated. The list is grouped by Added, Updated, and Deleted.

    b. To view the policies that include a specific check being updated, select the check in the list.

       A list of affected policies appears in the Related Policies pane.
4. (Optional) To view details about the policies affected:
    a. Click the **Policy Detail** tab.

       In the left pane is an alphabetical list of the policies affected by the update.

       > **Note:** The list shows only those policies that are affected by updated checks. The Policy Detail tab does not show other policy changes that could be included in the update, such as associating new checks with a policy or changing a policy name.

    b. To view the checks being updated in a specific policy, select the policy in the list.

       A list showing the ID, Name, and Version of checks being updated appears in the Related Checks pane. The list is grouped by Added, Updated, and Deleted.
5. To install the updates, click **Download**.

Downloading Checks without Updating Fortify WebInspect

Engine updates are required for some checks to be run during scans. If you are not using the latest version of Fortify WebInspect, it is likely that some of the checks in your SecureBase cannot be run during a scan. To test your application with all the latest checks, ensure that you are using the latest version of Fortify WebInspect.

Performing a SmartUpdate (Offline)

Follow this process to perform a SmartUpdate for WebInspect that is offline.

Stage	Description
1.	Open a support case. Customer Support personnel will provide you with the offline FTP server URL and login credentials (if needed). For more information, see Contacting Micro Focus Fortify Customer Support in "Preface" on page 19.
2.	On a machine that can access the Internet, access the offline FTP server.
3.	Download the Fortify WebInspect static SmartUpdate ZIP file.

Stage	Description
4.	On the machine where Fortify WebInspect is installed, extract all files from the ZIP file.
5.	Close Fortify WebInspect.
6.	Copy the extracted **SecureBase.sdf** and **version.txt** files to the directory where your SecureBase data resides. • If your system is not FIPS enabled, then the default location is C:\ProgramData\HP\HP WebInspect\SecureBase. • If your system is FIPS enabled, then the location is C:\ProgramData\HP\HP WebInspect\FIPS\SecureBase. **Note:** By default, these folders are hidden in Windows. Be sure to change folder options to show hidden files.

Chapter 12: About the SQL Injector Tool (Fortify WebInspect Only)

SQL injection is a technique for exploiting web applications that use client-supplied data in SQL queries without first removing potentially harmful characters. The SQL Injector supports MS-SQL, Oracle, Postgress, MySQL, and DB2 database types and also supports multiple language systems including Japanese.

Caution! This tool tests for SQL injection vulnerabilities by creating and submitting HTTP requests that may be processed by your SQL server. If your web application allows database records to be updated or created using data supplied by the user, the SQL Injector may create spurious records. To avoid this possibility, do not test against your production database. Instead, use a copy of the database, or use a test account that does not have access to the production data, or exclude from audit any pages that may update data or delete data from the database. If these alternatives are not feasible, back up your production database before testing at a time when the site has little or no customer traffic.

To test for susceptibility to SQL injection:

1. If using a proxy server or if the target site requires authentication, click the **Settings** tab and enter the appropriate information. For more information, see "SQL Injector Settings" on page 111.
2. Select **File > New**

   - or -

   click the New Request icon .
3. In the **Location** field, type or paste the URL that you suspect is susceptible to SQL injection. See examples below.
   - GET method (query parameters are embedded in the URL):

     http://172.16.61.10/Myweb/MSSQL/Welcome.asp?login=aaa&password=bbb
   - POST method (query parameters are included in message body):

     http://172.16.61.10:80/Myweb/MSSQL/Welcome.asp

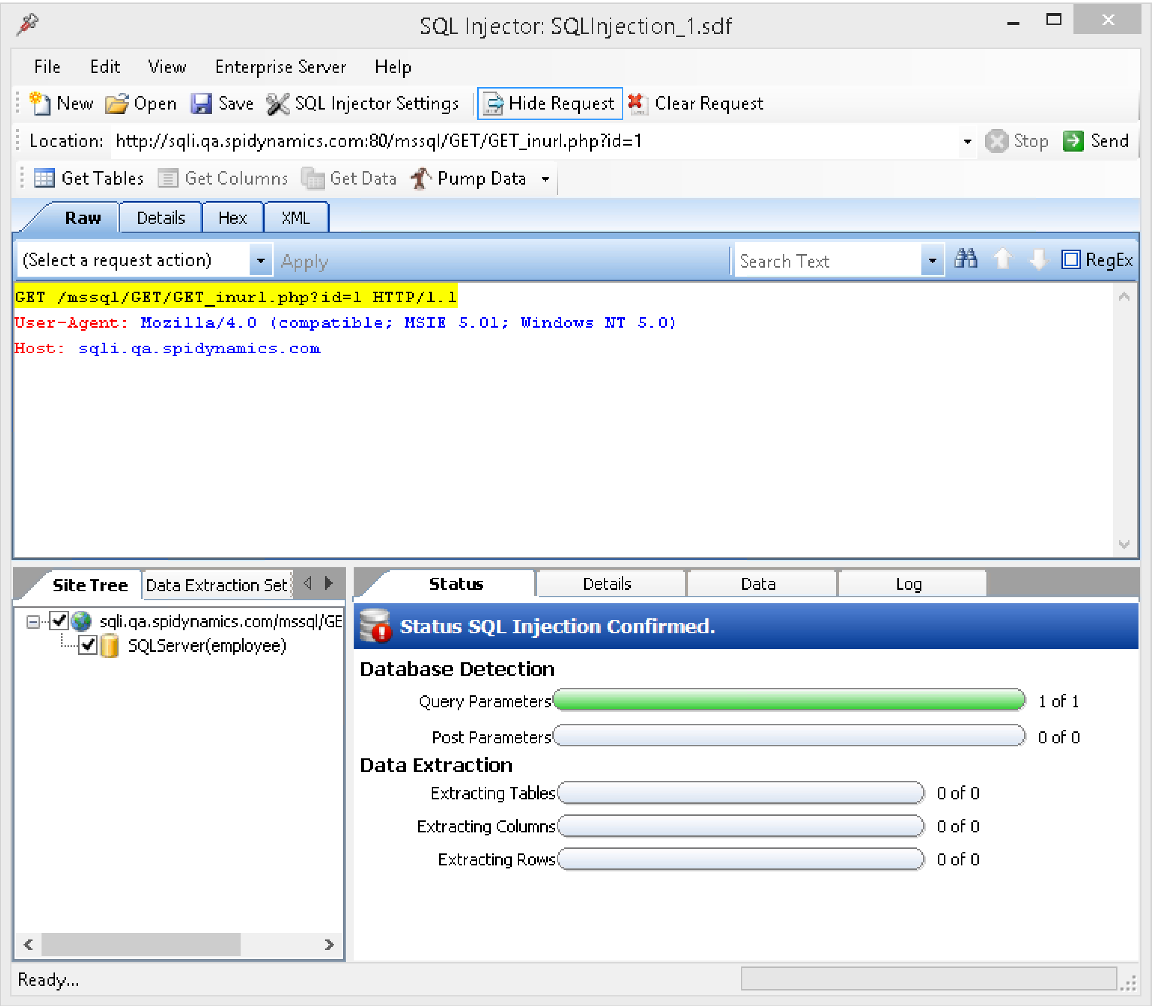
   Because the SQL Injector defaults to the GET method, you must also edit POST requests on the **Raw** tab (visible if you select **View > Show Request**). The edited request would be similar to the following:

   ```
   POST /Myweb/MSSQL/POST/2.asp HTTP/1.1
   User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
   Host: 172.16.61.10
   Content-Length: 22
   Content-Type: application/x-www-form-urlencoded
   login=qqq&password=aaa
   ```

> **Note:** If Fortify WebInspect has detected a SQL injection vulnerability, you can right-click the vulnerable session in Fortify WebInspect's navigation pane (or right-click the vulnerable URL on the **Vulnerabilities** tab of the summary pane) and select **Tools > SQL Injector** from the shortcut menu.

4. Click **Send**.

   If SQL injection is successful, "SQL Injection Confirmed" appears on the **Status** tab and the beginnings of a data hierarchy tree appear on the **Site Tree** tab in the lower left pane.



Location: http://sqli.qa.spidynamics.com:80/mssql/GET/GET_inurl.php?id=1

```
GET /mssql/GET/GET_inurl.php?id=1 HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
Host: sqli.qa.spidynamics.com
```

☑ sqli.qa.spidynamics.com/mssql/GE
☑ SQLServer(employee)

Status SQL Injection Confirmed.

Database Detection

Query Parameters	1 of 1
Post Parameters	0 of 0

Data Extraction

Extracting Tables	0 of 0
Extracting Columns	0 of 0
Extracting Rows	0 of 0

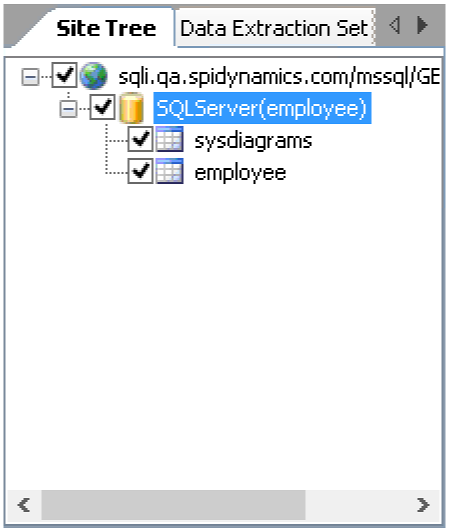
   For detailed information about the tabs on this screen, see "SQL Injector Tabs" on page 110.

5. To extract all the data from all tables, click the **Pump Data** icon Pump Data ▾.

   Alternatively, you can selectively investigate tables and columns using the following procedure:

   a. Select **Get Tables**.

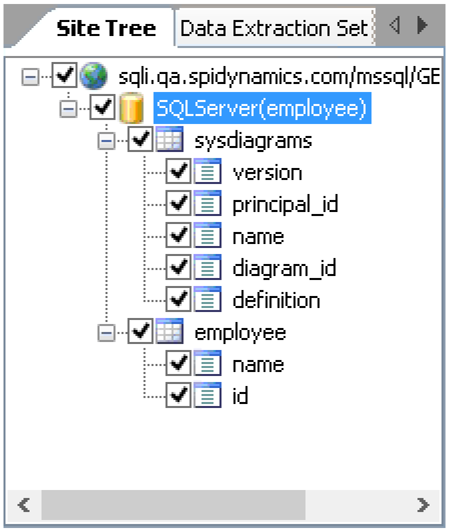
      The SQL Injector returns the names of all tables in the targeted database.



b. Choose tables by selecting or clearing their associated check boxes.

c. Click **Get Columns**.

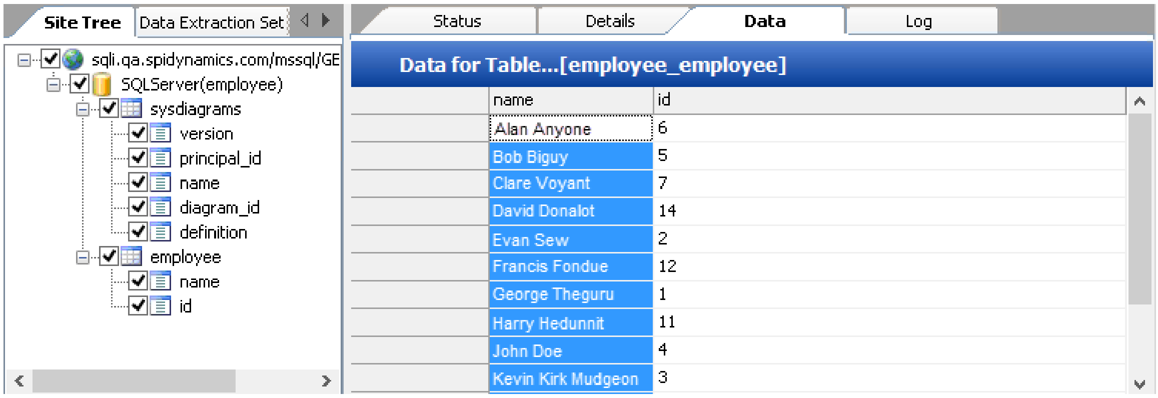
The SQL Injector returns the names of all columns in the selected tables.



d. Choose a column by selecting or clearing its associated check box.

e. Click **Get Data**.

6. Select a column and click the **Data** tab to view the column values.



Data for Table...[employee_employee]

name	id
Alan Anyone	6
Bob Biguy	5
Clare Voyant	7
David Donalot	14
Evan Sew	2
Francis Fondue	12
George Theguru	1
Harry Hedunnit	11
John Doe	4
Kevin Kirk Mudgeon	3

Note: If the SQL Injector is unable to extract data, it may be able to verify the existence of a SQL injection vulnerability by retrieving the name of the vulnerable database. To enable this feature, see Inferential/Time-Based Extraction in the "SQL Injector Settings" on the next page topic.

See Also

"SQL Injector Tabs" below

"SQL Injector Settings" on the next page

SQL Injector Tabs

After a successful SQL injection, the SQL Injector displays the following panes and tabs:

Request Pane

The Request pane contains the following tabs:

- **Raw** - Displays the text of the HTTP request.
- **Details** - Displays the request segmented by method, request URI, and protocol. Also lists the request header fields and their associated values.
- **Hex** - Displays a hexadecimal representation of the HTTP request.

To toggle the display of the Request pane, click **Show Request**/**Hide Request**.

To delete the request, replacing it with the default http://localhost:80/, click **Clear Request**.

Database Pane

The lower left pane contains the following tabs:

- **Site Tree** - Displays the URL, databases, tables, and columns.
- **Data Extraction Settings** - Displays the maximum number of tables, columns, and rows to return when extracting data. These values are extracted from the settings, but can be modified here or in the *Settings* dialog.

Information Pane

The lower right pane contains the following tabs:

- **Status** - Displays progress bars for detection and extraction functions.
- **Details** - Displays database information and injectable parameter details.
- **Data** - Displays data extracted from the selected tables and columns.
- **Log** - Displays a synopsis of pertinent functions and the time at which they occurred.

SQL Injector Settings

To modify the SQL Injector settings:

1. Click **Edit > Settings**.
2. Select one of the following tabs and specify settings as described in the following sections:
    - Options (See "Options Tab" below)
    - Authentication (See "Authentication Tab" on the next page)
    - Proxy (See "Proxy Tab" on page 113)
3. Click **OK**.

Options Tab

Timeout in Seconds

Specify the number of seconds that the SQL Injector will wait for a response before terminating the session.

Apply State

If your application uses cookies, URL rewriting, or post data techniques to maintain state within a session, the SQL Injector will attempt to identify the method and modify the response accordingly.

Apply Proxy

If you select this option, the SQL Injector will modify the request according to the proxy settings you specify.

Logging

Select the events you want to log:

- Requests
- Responses
- Errors
- Debug Messages

Log files are stored in xml format in `<drive>:\Users\<user name>\Documents\HP\Tools\SQLInjector\logs`.

The beginning of each file name is formatted as YYYY_MM_DD<current-process-id>. The remainder of the name is formatted as follows:

- _sqli_debug.log: Contains debugging messages for that session.
- _errors.log: Contains errors and exceptions that occurred for that session.
- _RequestsResponses.log: Contains all the requests and responses sent and received by the SQL Injector.

Data Extraction

Specify the maximum number of tables, columns, and rows that should be returned when extracting data through a URL that is vulnerable to SQL injection. These values are also displayed in the Database pane on the **Data Extraction Settings** tab. You can change these values using either this tab or the *Settings* dialog.

Also specify the maximum number of concurrent threads that should be used for data extraction.

Inferential/Time-Based Extraction

The SQL Injector can use two different techniques for extracting data when a SQL injection vulnerability is discovered. All attempts are conducted using the inferential technique, which examines the content of the HTTP responses. If this method fails, you can force the tool to use a second technique called time-based extraction. Instead of extracting table data, this method attempts to retrieve the name of the database by sending 4-5 long-running database queries for each character in the database name. Since this can be a rather time-consuming exercise, you can specify the number of characters required to confirm the existence of the SQL injection vulnerability.

Use a Macro

If you want to use a macro, select this check box and then click the browse button ... to select a macro.

Database File Path

This read-only text box displays the path to the database created by the SQL Injector tool to store attack data and replicate portions of the attacked database.

Authentication Tab

Authentication Method

If the site does not require authentication, select **None**. Otherwise, select an authentication method from the **Authentication** list:

- **Automatic** - If you select this option, the SQL Injector tool will determine the correct authentication method.

  > **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

- **HTTP Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.

- **NTLM** - NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

  Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and the scanner has to pass through a proxy server to submit its requests to the web server, the scanner may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

Authentication Credentials

Enter a user ID in the **User name** field and the user's password in the **Password** field. To prevent mistyping, repeat the password in the **Confirm Password** field.

Proxy Tab

Use these settings to access the SQL Injector through a proxy server.

Direct Connection (proxy disabled)

Select this option if you are not using a proxy server.

Auto detect proxy settings

If you select this option, SQL Injector will use the Web Proxy Autodiscovery Protocol (WPAD) to locate and use a proxy autoconfig file to configure the browser's web proxy settings.

Use System proxy settings

Select this option to import your proxy server information from the local machine.

Use Firefox proxy settings

Select this option to import your proxy server information from Firefox.

> **Note:** Using browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.

Configure a proxy using a PAC file

Select this option to load proxy settings from the Proxy Automatic Configuration (PAC) file in the file location you specify in the **URL** field.

Explicitly configure proxy

Select this option to access the Internet through a proxy server, and then enter the requested information:

1. In the **Server** field, type the URL or IP address of your proxy server, followed (in the **Port** field) by the port number (for example, 8080).
2. Select a protocol **Type** for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or Standard.
3. If authentication is required, select a type from the **Authentication** list:
   - **Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

     The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.

   - **NTLM** - NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

     Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and SQL Injector has to pass through a proxy server to submit its requests to the web server, SQL Injector may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

   - **Kerberos** - Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

   - **Digest** - The Windows Server operating system implements the Digest authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using Digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

- **Automatic** - Allow SQL Injector to determine the correct authentication type.

> **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

- **Negotiate** - The Negotiate authentication protocol begins with the option to negotiate for an authentication protocol. When the client requests access to a service, the server replies with a list of authentication protocols that it can support and an authentication challenge based on the protocol that is its first choice. For example, the server might list Kerberos and NTLM, and send a Kerberos challenge. The client examines the contents of the reply and checks to determine whether it supports any of the specified protocols. If the client supports the preferred protocol, authentication proceeds. If the client does not support the preferred protocol, but does support one of the other protocols listed by the server, the client lets the server know which authentication protocol it supports, and the authentication proceeds. If the client does not support any of the listed protocols, the authentication exchange fails.

4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** field. Use commas to separate entries.

Specify Alternative Proxy for HTTPS

For proxy servers accepting HTTPS connections, select the **Specify Alternative Proxy for HTTPS** check box and provide the requested information.

Chapter 13: About the SWFScan Tool (Fortify WebInspect Only)

The SWFScan tool can help organizations secure applications developed using the Adobe Flash platform. This innovative tool identifies many of the vulnerabilities that affect Flash applications and provides definitive insight on how to remove or avoid them. For more information about the vulnerabilities, see "Vulnerability Detection" below.

How It Works

SWFScan uniquely supports all versions of Adobe Flash and ActionScript, including ActionScript 2 and 3 (Flash versions 9 and 10).

When you point SWFScan at a Flash file on the Internet or intranet, or load a Flash file from your local computer, SWFScan decompiles the SWF bytecode, generates ActionScript source code, and performs static analysis. You can then generate reports that include:

- Identification of the source code that caused the vulnerability
- Implications of each specific vulnerability
- "Best practice" guidelines to help with remediation

SWFScan also provides additional key information (such as networking calls, external domain requests, etc.) that may be useful for manual inspection of your Flash applications.

See Also

"Analyzing Flash Files" on the next page

Vulnerability Detection

SWFScan tests for the following types of Flash security vulnerabilities. Checks for additional vulnerabilities will be added to SecureBase (through SmartUpdate) as they are developed.

ActionScript 3 Vulnerabilities Detected by SWFScan

SWFScan finds the following types of vulnerabilities in applications built on Flash 9 and above:

- Insecure Programming Practice
- Insecure Application Deployment
- Adobe Best Practices Violation
- Information Disclosure

For information on specific checks, select **Edit > Settings**, and then select the **Checks** tab.

ActionScript 1 and 2 Vulnerabilities Detected by SWFScan

SWFScan finds the following types of vulnerabilities in applications built on Flash 8 and below:

- Possible Cross-Site Scripting
- Dangerous functions accepting user supplied data
- Insecure Programming Practice
- Insecure Application Deployment
- Information Disclosure

For information on specific checks, select **Edit > Settings**, and then select the **Checks** tab.

Analyzing Flash Files

You can use SWFScan as a standalone tool or as an integrated component of Fortify WebInspect to analyze Flash files.

Using SWFScan as a Standalone Tool

To analyze a Flash file using SWFScan as a standalone tool:

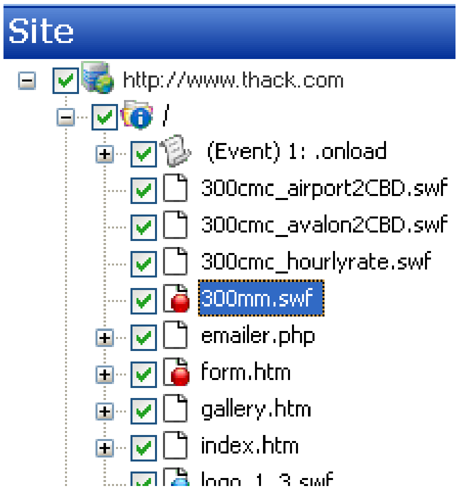
1. To launch SWFScan, click **Start > All Programs > Fortify > Fortify Security Toolkit > SwfScan**.
2. Specify the Flash file (.swf) you want to analyze. Do one of the following:
   - In the **Path or URL** combo box, enter or select the full path to a Flash file and click Get on the SWFScan toolbar.
   - Click **File > Open**, select a Flash file from a local storage device, and click **Open**.

   SWFScan loads and decompiles the selected file.
3. Click Analyze on the SWFScan toolbar.

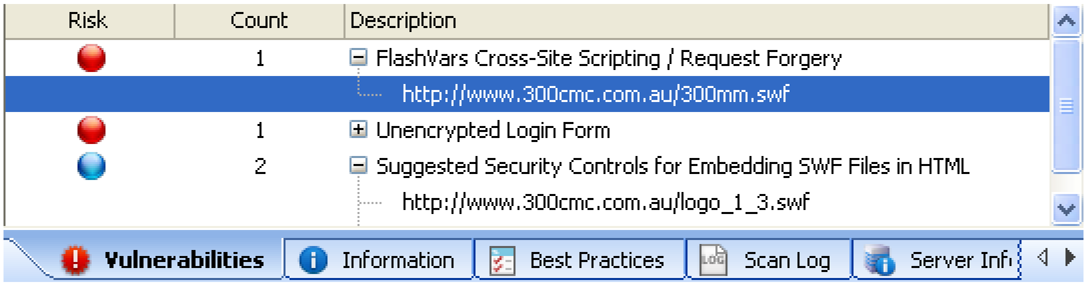
Using SWFScan in Fortify WebInspect

To analyze a Flash file using SWFScan as an integrated component of Fortify WebInspect:

1. Do one of the following while or after conducting a scan:
   - Locate a Flash file (.swf) in the navigation pane, then right-click the file name and select **Tools > SWFScan** from the shortcut menu.



- Locate a Flash vulnerability on the **Vulnerabilities** tab, then right-click an associated URL and select **Tools > SWFScan** from the shortcut menu.



Risk	Count	Description
●	1	⊟ FlashVars Cross-Site Scripting / Request Forgery
		http://www.300cmc.com.au/300mm.swf
●	1	⊞ Unencrypted Login Form
●	2	⊟ Suggested Security Controls for Embedding SWF Files in HTML
		http://www.300cmc.com.au/logo_1_3.swf

The SWFScan tool launches and loads the decompiled source code.

2. Click Analyze on the SWFScan toolbar.

Note: Fortify WebInspect analyzes Flash files if this function is enabled in the Default settings (located in **Scan Settings > Content Analyzers**). However, SWFScan offers more functionality and control by allowing you to configure independent settings, export source code and discovered URLs, and generate individual reports for each file. You can also search the source code or specific portions of it (see "Searching Source Code" on the next page).

See Also

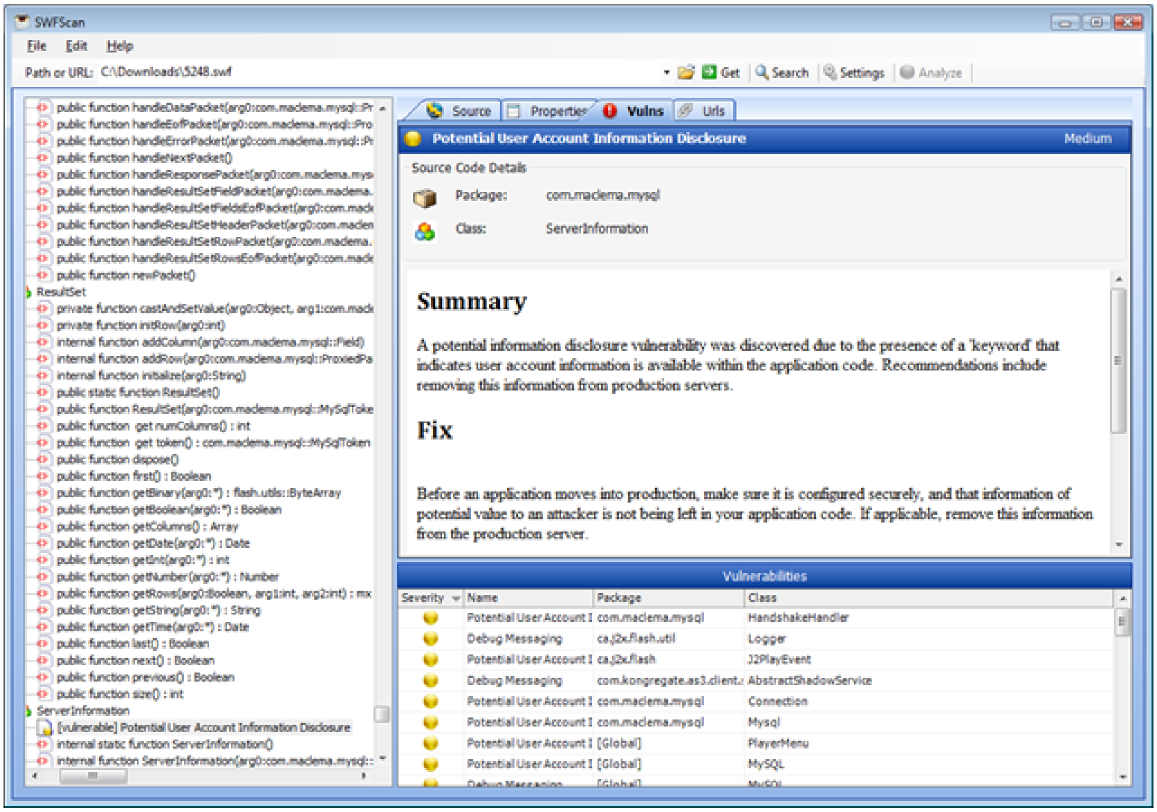
"Examining Results" below

"Searching Source Code" on the next page

Examining Results

SWFScan displays a list of detected vulnerabilities in the lower right pane.

Click an item in the list to display information about the vulnerability and to locate (in the left pane) the module in which the vulnerability was detected.



Path or URL: C:\Downloads\5248.swf

```
public function handleDataPacket(arg0:com.maclema.mysql::Pr
public function handleEofPacket(arg0:com.maclema.mysql::Pro
public function handleErrorPacket(arg0:com.maclema.mysql::Pr
public function handleNextPacket()
public function handleResponsePacket(arg0:com.maclema.mys
public function handleResultSetFieldPacket(arg0:com.maclema.
public function handleResultSetFieldsEofPacket(arg0:com.macl
public function handleResultSetHeaderPacket(arg0:com.maclem
public function handleResultSetRowPacket(arg0:com.maclema.
public function handleResultSetRowsEofPacket(arg0:com.macl
public function newPacket()
ResultSet
private function castAndSetValue(arg0:Object, arg1:com.macl
private function initRow(arg0:int)
internal function addColumn(arg0:com.maclema.mysql::Field)
internal function addRow(arg0:com.maclema.mysql::ProxiedPa
internal function initialize(arg0:String)
public static function ResultSet()
public function ResultSet(arg0:com.maclema.mysql::MySqlToke
public function  get numColumns() : int
public function  get token() : com.maclema.mysql::MySqlToken
public function dispose()
public function first() : Boolean
public function getBinary(arg0:*) : flash.utils::ByteArray
public function getBoolean(arg0:*) : Boolean
public function getColumns() : Array
public function getDate(arg0:*) : Date
public function getInt(arg0:*) : int
public function getNumber(arg0:*) : Number
public function getRows(arg0:Boolean, arg1:int, arg2:int) : mx
public function getString(arg0:*) : String
public function getTime(arg0:*) : Date
public function last() : Boolean
public function next() : Boolean
public function previous() : Boolean
public function size() : int
ServerInformation
[vulnerable] Potential User Account Information Disclosure
internal static function ServerInformation()
internal function ServerInformation(arg0:com.maclema.mysql::
```

Potential User Account Information Disclosure — Medium

Source Code Details

Package: com.maclema.mysql

Class: ServerInformation

Summary

A potential information disclosure vulnerability was discovered due to the presence of a 'keyword' that indicates user account information is available within the application code. Recommendations include removing this information from production servers.

Fix

Before an application moves into production, make sure it is configured securely, and that information of potential value to an attacker is not being left in your application code. If applicable, remove this information from the production server.

Vulnerabilities

Severity	Name	Package	Class
●	Potential User Account I	com.maclema.mysql	HandshakeHandler
●	Debug Messaging	ca.j2x.flash.util	Logger
●	Potential User Account I	ca.j2x.flash	J2PlayEvent
●	Debug Messaging	com.kongregate.as3.client.	AbstractShadowService
●	Potential User Account I	com.maclema.mysql	Connection
●	Potential User Account I	com.maclema.mysql	Mysql
●	Potential User Account I	[Global]	PlayerMenu
●	Potential User Account I	[Global]	MySQL
●	Debug Messaging	[Global]	MySQL

Searching Source Code

You can search for specific text strings or text strings that match the regular expression you specify.

1. In the **Search For** field, enter a text string or regular expression.
2. To find only those occurrences matching the case of the text string or regular expression, select the **Match Case** check box.
3. To identify the string as a regular expression, select **RegEx**.
4. Choose the specific area that you want to search.

   For ActionScript 2 files:

   - All Source Code—The decompiled source code.
   - Specific Movie Clip—Select a clip from the list.
   - Specific Frame—Select a clip and a frame.
   - Specific Class—Select a class from the list.
   - Specific Method—Select a class and a method.

   For ActionScript 3 files:

   - All Source Code—The decompiled source code.
   - Specific Package—Select a package from the list.

- Specific Class—Select a package and class.
- Specific Method—Select a package, class, and method.

5. Click **Search**.

The results appear on the **Search Results** tab, with matches highlighted.

Configuring SWFScan Settings

To configure SWFScan settings:

1. Click on the SWFScan toolbar.
2. Configure settings on the various tabs. For more information about the settings on each tab, see the following topics:

   "AS2 Exclusions" below

   "AS3 Exclusions" on the next page

   "Proxy" on the next page

   "Checks" on page 123
3. Click **OK**.

Changed settings are persisted, but cannot be applied retroactively. To analyze a Flash file after changing settings, you must click .

AS2 Exclusions

You can exclude ActionScript 2 packages (namespaces) from analysis by selecting the **Enabled** check box associated with a particular package.

Clear the check box if you want to include the package in your analysis.

To add an exclusion to the list:

1. Click **Add**.
2. On the Add Exclusion Rule window, enter a name for the rule and a regular expression that describes the package.
3. Click **OK**.

You can also edit or remove any rules that you add, but you cannot modify the default rule (the Flash Standard Library).

AS3 Exclusions

You can exclude ActionScript 3 packages (namespaces and classes) from analysis by selecting the **Enabled** check box associated with a particular package or class.

Clear the check box if you want to include the package or class in your analysis.

To add packages and classes to the exclusion list:

1. Click **Add**.
2. On the Add Exclusion Rule window, enter a name for the rule and a regular expression that describes the package or class.
3. Click **OK**.

You can also edit or remove any rules that you add, but you cannot modify the default rules.

Proxy

Select from the following options:

- **Direct Connection (proxy disabled)** - Select this option if you are not using a proxy server.
- **Auto detect proxy settings** - If you select this option, the SWFScan tool will use the Web Proxy Autodiscovery (WPAD) protocol to locate and use a proxy autoconfig file to configure the browser's web proxy settings.
- **Use System proxy settings** - Select this option to import your proxy server information from the local machine.
- **Use Firefox proxy settings** - Select this option to import your proxy server information from Firefox.

> **Note:** Using browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.

- **Configure a proxy using a PAC file** - Select this option to load proxy settings from a Proxy Automatic Configuration (PAC) file. Then specify the file location in the **URL** field.
- **Explicitly configure proxy** - Select this option to access the Internet through a proxy server, and then enter the requested information:
  a. In the **Server** field, type the URL or IP address of your proxy server, followed (in the **Port** field) by the port number (for example, 8080).
  b. Select a protocol **Type** for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or Standard.
  c. If authentication is required, select a type from the **Authentication** list:
     - **Basic** - A widely used, industry-standard method for collecting user name and password information. The web browser displays a window for a user to enter a previously assigned user

name and password, also known as credentials. The web browser then attempts to establish a connection to a server using the user's credentials. If the user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the web server verifies that the user name and password correspond to a valid user account, a connection is established.

The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your web server is secure.

- **NTLM** - NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

  Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and the scanner has to pass through a proxy server to submit its requests to the web server, the scanner may not be able to crawl or audit that website. Use caution when configuring this tool for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

- **Kerberos** - Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

- **Digest** - The Windows Server operating system implements the Digest authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using Digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

- **Automatic** - Allow SWFScan to determine the correct authentication type.

  > **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

- **Negotiate** - The Negotiate authentication protocol begins with the option to negotiate for an authentication protocol. When the client requests access to a service, the server replies with a list of authentication protocols that it can support and an authentication challenge based on the protocol that is its first choice. For example, the server might list Kerberos and NTLM, and send a Kerberos challenge. The client examines the contents of the reply and checks to determine whether it supports any of the specified protocols. If the client supports the preferred protocol, authentication proceeds. If the client does not support the preferred protocol, but does support one of the other protocols listed by the server, the client lets the server know which authentication protocol it supports, and the authentication proceeds. If the client does not support any of the listed protocols, the authentication exchange fails.

d. If your proxy server requires authentication, enter the qualifying **User name** and **Password**.

e. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** field. Use commas to separate entries.

- **Specify Alternative Proxy for HTTPS** - For proxy servers accepting HTTPS connections, select the **Specify Alternative Proxy for HTTPS** check box and provide the requested information.

Checks

This tab lists all attacks that check for specific vulnerabilities in the decompiled code.

Sorting the Checks

By default, the list of checks is sorted by **Severity**, from Critical down to Best Practice. To sort alphabetically by **Check Name**, click the column heading. To reverse the sort order of the selected column, click the column heading again.

Enabling/Disabling Checks

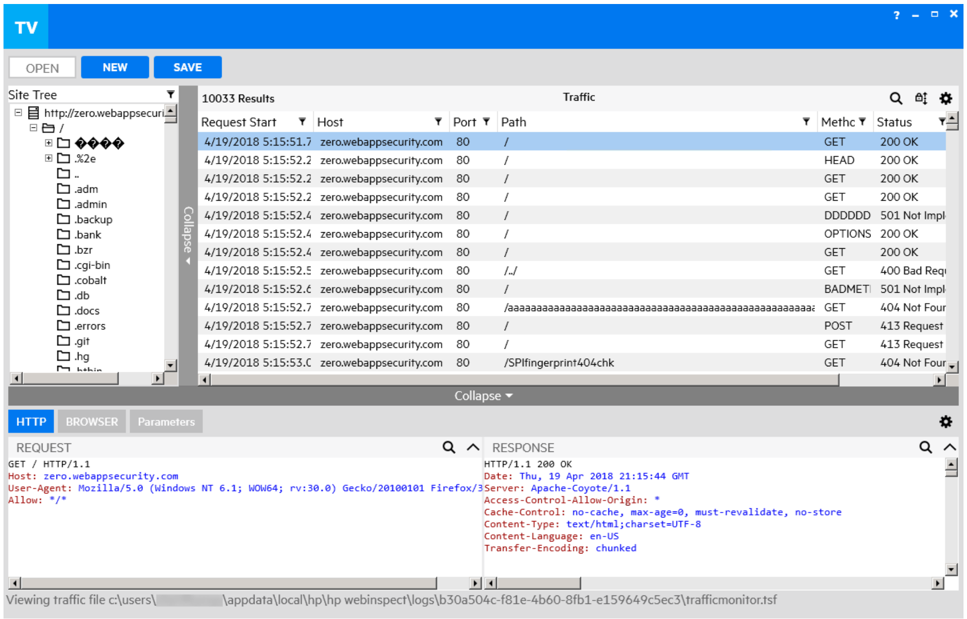
By default, all checks in the list are enabled, as indicated by the selected check boxes in the **Enabled** column. You can disable a check by clearing its check box in the **Enabled** column.

Chapter 14: About the Traffic Viewer

Fortify WebInspect normally displays in the navigation pane only the hierarchical structure of the Web site or Web service, plus those sessions in which a vulnerability was discovered. The Traffic Viewer, however, allows you to display and review every HTTP request sent by Fortify WebInspect and the associated HTTP response received from the server.

Traffic Viewer Image

The following image shows the Traffic Viewer displaying a traffic file from a scan.



10033 Results

Request Start	Host	Port	Path	Methc	Status
4/19/2018 5:15:51.7	zero.webappsecurity.com	80	/	GET	200 OK
4/19/2018 5:15:52.2	zero.webappsecurity.com	80	/	HEAD	200 OK
4/19/2018 5:15:52.2	zero.webappsecurity.com	80	/	GET	200 OK
4/19/2018 5:15:52.2	zero.webappsecurity.com	80	/	GET	200 OK
4/19/2018 5:15:52.4	zero.webappsecurity.com	80	/	DDDDDD	501 Not Impl
4/19/2018 5:15:52.4	zero.webappsecurity.com	80	/	OPTIONS	200 OK
4/19/2018 5:15:52.4	zero.webappsecurity.com	80	/	GET	200 OK
4/19/2018 5:15:52.5	zero.webappsecurity.com	80	/../	GET	400 Bad Requ
4/19/2018 5:15:52.6	zero.webappsecurity.com	80	/	BADMETI	501 Not Impl
4/19/2018 5:15:52.7	zero.webappsecurity.com	80	/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa	GET	404 Not Four
4/19/2018 5:15:52.7	zero.webappsecurity.com	80	/	POST	413 Request
4/19/2018 5:15:52.7	zero.webappsecurity.com	80	/	GET	413 Request
4/19/2018 5:15:53.0	zero.webappsecurity.com	80	/SPIfingerprint404chk	GET	404 Not Four

```
GET / HTTP/1.1
Host: zero.webappsecurity.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:30.0) Gecko/20100101 Firefox/3
Allow: */*
```

```
HTTP/1.1 200 OK
Date: Thu, 19 Apr 2018 21:15:44 GMT
Server: Apache-Coyote/1.1
Access-Control-Allow-Origin: *
Cache-Control: no-cache, max-age=0, must-revalidate, no-store
Content-Type: text/html;charset=UTF-8
Content-Language: en-US
Transfer-Encoding: chunked
```

Viewing traffic file c:\users\[redacted]\appdata\local\hp\hp webinspect\logs\b30a504c-f81e-4b60-8fb1-e159649c5ec3\trafficmonitor.tsf

Option Must be Enabled

To use the Traffic Viewer, you must enable the Traffic Monitor Logging option prior to running a scan. The Traffic Viewer is not available for a scan if the Traffic Monitor Logging option was not enabled prior to running the scan. See "Enabling Traffic Monitor" on the next page for more information.

Proxy Server

The Traffic Viewer also includes a self-contained proxy server that you can configure and run on your desktop. With it, you can monitor traffic from your browser as it submits HTTP requests and receives responses from a Web server. The Traffic Viewer proxy is a tool for debugging and penetration assessment; you can see every request and server response while browsing a site.

You can also use this feature to create a Workflow macro or a Login macro that you can use with Fortify WebInspect.

Enabling Traffic Monitor

In Fortify WebInspect, you can enable the Traffic Monitor for all scans or for an individual scan.

Enabling the Traffic Monitor for All Scans

To enable the Traffic Monitor in the Default Settings:

1. Click **Edit > Default Scan Settings**.
2. In the Scan Settings pane, click **General**.
3. Select **Enable Traffic Monitor Logging**.

   > **Note:** The Traffic Viewer does not support the encryption of traffic files. The **Encrypt Traffic Monitor File** option is for use under special circumstances with legacy traffic files only.

4. Click **OK**.

Enabling the Traffic Monitor for Individual Scans

To enable the Traffic Monitor when you start a scan through the Scan Wizard, do one of the following:

- Select **Settings (Default)** at the bottom of the Scan Wizard and follow steps 2 through 4 of "Enabling the Traffic Monitor for All Scans" above.
- In the Detailed Scan Configuration window of the Scan Wizard, select **Enable Traffic Monitor**.

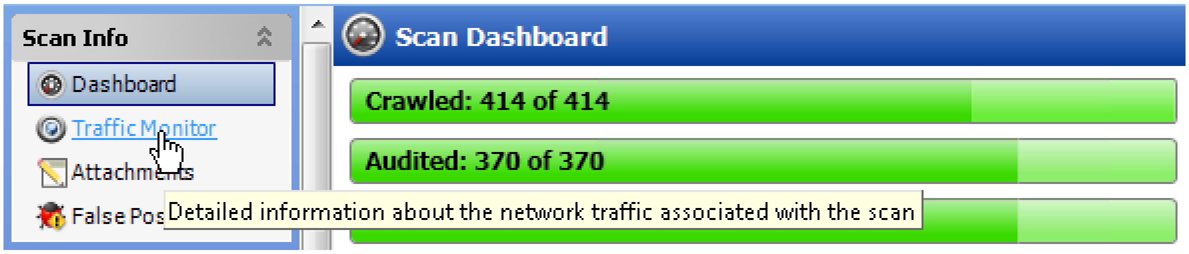
Launching the Traffic Viewer

You can launch the Traffic Viewer from the Scan Info Panel within an open scan in Fortify WebInspect and Fortify WebInspect Enterprise. Launching the tool in this manner opens the Traffic Viewer with a traffic file in view. You can also open the tool as a stand-alone tool outside of a scan and without any traffic or proxy data in view.

From an Open Scan

To launch the Traffic Viewer from an open scan in Fortify WebInspect and Fortify WebInspect Enterprise:

- In the Scan Info panel, click **Traffic Monitor**



Crawled: 414 of 414

Audited: 370 of 370

Detailed information about the network traffic associated with the scan

> **Note:** The Traffic Viewer is not available if Traffic Monitor Logging was not enabled prior to conducting the scan.

As a Stand-alone Tool

To launch the stand alone Traffic Viewer, do one of the following:

- In Fortify WebInspect, click **Tools > Traffic Viewer**.
- In the Fortify WebInspect Enterprise Admin Console, click **Tools > Traffic Viewer**.

The Traffic Viewer is launched without any traffic or proxy data in view.

> **Note:** You may also launch the Traffic Viewer from your Windows Start menu.

Using the Site Tree

By default, the Site Tree displays an unfiltered tree view of all traffic that was generated during the scan. The tree includes a list of hosts and all sub-directories within those hosts. In this view, you can select a top-level host and expand the sub-directories to examine the requests and responses occurring at each level. You can select an item in the Site Tree to display the traffic for the item.

Site Tree Icons

The following table identifies the icons displayed in the Site Tree.

Icon	Name	Represents
▤	Server/host	The top level of your site's tree structure
🗀/🗁	Folder	A directory

Icon	Name	Represents
🗋	Page	A file

Viewing Traffic for a Resource

You can view the traffic for a resource in the Site Tree. To view the traffic for an item:

- Select the item in the **Site Tree**.

  All traffic involving that item appears in the Traffic grid.

For more information, see "Working with Sessions" on page 132.

Viewing Only Host Names

To view a list of only the host names:

- From the default tree view, click the filter icon once.

  The Site Tree displays only the host names. Sub-directories are not accessible in this view. From this view, you can select one or more hosts and filter out the rest. See "Filtering for Selected Hosts" below.

To return to viewing the entire tree:

- Click the filter icon again.

Filtering for Selected Hosts

To focus your research, you can filter for specific hosts in the Site Tree. To view only selected hosts and their sub-directories in the Site Tree:

1. With the Site Tree displaying only the host names, select one or more hosts to view.
2. Click the filter icon.

   Only the selected hosts appear in the Site Tree.
3. Expand a host to display its sub-directories.

Viewing All Host Names

To return to viewing all host names:

1. Click the filter icon.

   The Site Tree displays only the host names with the previously viewed hosts selected.
2. Click each selected host to clear its selection.
3. Click the filter icon.

   The Site Tree displays an unfiltered tree view of all traffic.

See Also

"Resizing, Collapsing, and Expanding UI Elements" on page 130

Customizing Grid Views

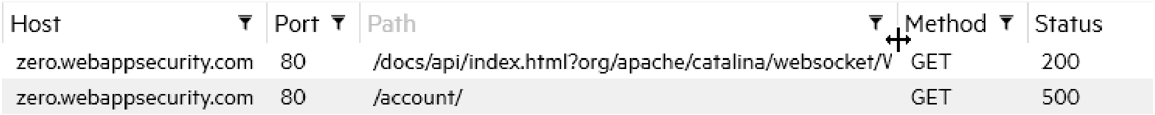
You can resize, reposition, add, and remove columns displayed in grid views.

Resizing Columns

To resize a column:

1. Move your cursor to the border to the right of the column heading you want to resize.

   Your cursor becomes a double-headed arrow and the column heading background color changes to a lighter gray.



Host	Port	Path	Method	Status
zero.webappsecurity.com	80	/docs/api/index.html?org/apache/catalina/websocket/V	GET	200
zero.webappsecurity.com	80	/account/	GET	500

2. Do one of the following:
   - Drag the column border either right or left to the width you want.
   - Double-click the border to resize the column to the width of the widest amount of data in the column. A horizontal scroll bar might be added to the bottom of the window.

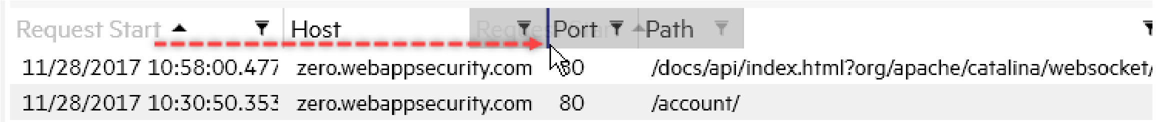
Repositioning Columns

To rearrange the order of the columns across the grid:

1. Move your cursor to the column heading that you want to move.

   The column heading background color changes to a lighter gray.
2. Click once.

   The column heading background color changes to white.
3. Drag the column to the right or left into the position you want it.



Request Start	Host	Port	Path
11/28/2017 10:58:00.477	zero.webappsecurity.com	80	/docs/api/index.html?org/apache/catalina/websocket/
11/28/2017 10:30:50.353	zero.webappsecurity.com	80	/account/

   The column of data is moved and the remaining columns are shifted right or left by one column.

Adding/Removing Columns

By default, not all columns of data are displayed in the grid. Grid view settings allow you to select which columns of data you want visible in the grid. To add or remove displayed columns:

1. In the grid view, click ⚙.

   A list of available columns appears.

   > **Note:** The column names indicate the memo headers that are generated during a scan.

2. Do the following:
   - Select the check box for each column you want to add to the display.
   - Clear the check box for each column you want to remove from the display.
3. Click anywhere outside the list of columns to close the list.
   The displayed columns are updated.

Customizing Detail Views

You can choose the layout and color theme for non-grid detail views, and you can hide or show the HTTP detail views.

Changing the Layout

When two detail views are visible for an item, such as the Request and Response detail views, you can rearrange the placement of the detail views to have them stacked vertically (one on top of the other) or have them aligned horizontally (side-by-side). To change the layout:

1. In the detail view, click ⚙.
   The settings menu opens.
2. Do one of the following:
   - To align the detail views vertically one on top of the other, click **Vertical Layout**.
   - To align the detail views horizontally side-by-side, click **Horizontal Layout**.

Changing the Color Theme

The default color theme is black and colored text on a white background. However, you might prefer white and colored text on a black background. To change the color theme:

1. In the detail view, click ⚙.
2. Do one of the following:
   - To use black and colored text on a white background, click **Light Theme**.
   - To use white and colored text on a black background, click **Dark Theme**.

Hiding and Showing HTTP Detail Views

You can collapse (or hide) one of the HTTP detail views, such as the Request or Response detail view, so that only the contents of other HTTP detail view is visible.

To hide a detail view:

- Click the hide icon (∧) in the detail view.

To show a hidden detail view:

- Click the show icon (∨).

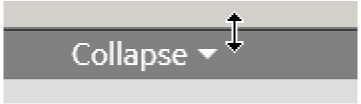
Resizing, Collapsing, and Expanding UI Elements

You can resize, hide (or collapse), and show (or expand) certain user interface (UI) elements, such as a site tree or a grid view of data.

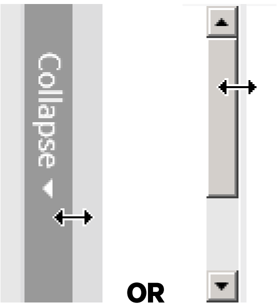
Resizing an Element

To resize an element, do one of the following:

- For UI elements with a horizontal layout of data, such as a grid view, drag the horizontal Collapse bar to widen or narrow the element.



- For UI elements with a vertical layout of data, such as a site tree, drag the vertical Collapse bar or the scroll bar to widen or narrow the panel.



Collapsing an Element

To collapse an element:

- Click **Collapse**.

Expanding an Element

To expand an element:

- Click **Expand**.

Using Auto Scroll

Enabling auto scroll causes the traffic grid to scroll up as new sessions are added so that the newest traffic sessions are always visible. The auto scroll feature is only applicable when you are working with a scan that is currently running.

Enabling Auto Scroll

To enable auto scroll:

- Click the scroll lock icon (🔒).

Disabling Auto Scroll

You may want to pause auto scroll to examine a session in the Traffic grid. To disable auto scroll:

- Click the scroll lock icon (🔒).

> **Note:** You can resume auto scroll at any time during the active scan.

Exploring Traffic

By default, the Traffic grid displays all traffic generated during the scan, allowing you to explore the traffic for the entire scan. However, you can also view and explore traffic for a specific resource. You can search, sort, and filter the data in the Traffic grid. For more information, see "Searching and Filtering" on page 136.

Viewing Traffic for a Resource

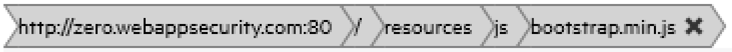
You can view the traffic for a resource in the Site Tree. To view the traffic for an item:

- Select the item in the **Site Tree**.

  All traffic involving that item appears in the Traffic grid.

Using the Breadcrumbs

When you select a resource in the Site Tree, breadcrumbs appear at the top of the traffic grid, similar to the sample shown here.



http://zero.webappsecurity.com:80 / resources js bootstrap.min.js ✖

These breadcrumbs indicate that the displayed traffic has been filtered down to the last resource listed in the breadcrumbs.

To filter the traffic for a specific resource listed elsewhere in the breadcrumbs:

- Click the resource in the breadcrumbs.

  > For example, if you want to view all traffic for the resources folder shown in the previous image, click **resources**.

  The selected resource becomes the final breadcrumb and the traffic sessions are updated to show only the traffic for the selected resource.

To remove the filter completely:

- Click **X** in the final breadcrumb.

  The breadcrumbs are removed and the traffic sessions are no longer filtered.

See Also

"Working with Sessions" below

"Drilling Down Into Traffic Data" on page 135

Working with Sessions

You cannot modify data you are viewing in a traffic file from a scan. You can, however, research the traffic data in the Traffic Viewer to get a better understanding of what happened during the scan. For example, you can resend a request using the HTTP Editor or you can view the session in a browser.

Viewing the HTTP Detail

You can view the request and response of a session in the HTTP detail view. This view is the default view for sessions selected in most grids. However, if you are seeing another detail view and want to see the request and response instead, you can switch to the HTTP detail view. To view a session in the HTTP detail view:

1. Select a session in the grid.
2. Click **HTTP**.

   The HTTP detail view opens, showing the request and response of the selected session.

Wrapping Text

Long lines of text in the detail views, such as in the Request and Response detail views, might make it impossible to view the content without using the horizontal scroll bars. You can use the Word Wrap setting to wrap the text to prevent the horizontal scroll bars. The Word Wrap setting is available in each detail view and is not a global setting for all detail views. The Word Wrap setting is saved in your user settings file for each detail view, and is the default behavior for the detail view the next time you open the application.

To wrap text:

- Right-click the detail view and select **Word Wrap**.

  The long lines of text are wrapped and the horizontal scroll bar is removed.

Decoding Percent-encoded Characters

By default, requests and responses use percent-encoding for reserved characters. If you see percent-encoded characters, such as `%3B` and `%40`, in the text of a request or response, you can decode these characters to improve readability of the text. When you decode the characters in a request or response, the requests or responses for all parent and child sessions of the selected session will also be decoded. These characters remain decoded only while the scan is open. If you close the scan and reopen it, the default display applies, and reserved characters will once again be percent-encoded.

To decode percent-encoded characters:

- Right-click in the **RESPONSE** or **REQUEST** tab and select **URL Decode**.

  The percent-encoded characters are converted to readable text.

Resending a Request

You can resend a request using the HTTP Editor. To resend a request:

1. Select a session in the grid to view the request and response.
2. If the HTTP detail view is not open, click **HTTP**.
3. Right-click in the **REQUEST** detail view and select **View in HTTP Editor**.

   The HTTP Editor opens for the request. For more information about using the HTTP Editor, see the HTTP Editor online help or the *Tools Guide for Fortify WebInspect Products*.

Viewing a Session in the Browser

You can view a session in the Browser detail view to see where the traffic occurred in your site. To view a session in the Browser:

1. Select a session in the grid.
2. Click **BROWSER**.

   The Browser detail view opens showing the selected session.

Expanding Compressed Content

Compressing (or minifying) content removes spaces, new line markers, comments, and block delimiters from code to reduce file size. However, the practice also makes the content more difficult for humans to read. You can use the Beautify setting to expand compressed text. The Beautify setting is available in each detail view and is not a global setting for all detail views. The Beautify setting is saved in your user settings file for each detail view, and is used as the default behavior for the detail view the next time you open the application.

To expand compressed content:

- Right-click in the detail view and select **Beautify**.

  The compressed content is expanded and becomes more readable.

  Note: Some text cannot be beautified, so you might not see the option.

See Also

"Working with Parameters" below

Working with Parameters

You can view the Type, Name, and Value for parameters used in a traffic session. The Parameters detail view displays a grid with one record for each cookie or query string used in the traffic session. You can also view every traffic record in which the same parameter is used. You can access the Parameters detail view from the Traffic and Related Traffic grids.

Understanding Parameters

A parameter can be one of the following:

- Cookie data
- A query string submitted as part of the URL in the HTTP request (or contained in another header)
- Data submitted using the Post method (such as set_<parametername>)

Viewing Parameter Details

To view the parameter details for a session:

1. Select a session in the Traffic or Related Traffic grid.
2. Click **PARAMETERS**.

The Parameters detail view opens showing the parameters used in the selected session.

> **Note:** The detail view layout settings have no effect on the Parameters grid.

Adding Parameter Columns to Traffic Grid

You can add columns to the Traffic grid to display a parameter that is listed in the Parameters detail view. Adding these columns of data to the Traffic grid is useful when you are working with a workflow macro and need to follow a state parameter through the sessions to determine when and why you are being logged out of the application.

> For example, you might want to view the values for the JSESSIONID parameter to examine it from session to session to see where its value changes. You can add a column for the JSESSIONID parameter along with its companion column set_JSESSIONID to show where the value changes.

To add columns for a parameter:

1. Right-click the row for the parameter in the Parameters detail grid.
2. Select **Build Columns...**.

   > **Note:** If you have previously added columns for the selected parameter, the Build Columns option is unavailable.

A column for the parameter name is added to the Traffic grid, along with a column for any methods that set the parameter value, if applicable. These columns are permanently added to the database for the current scan. The column names are also added to the grid settings menu. You can use the grid settings menu to add or remove the columns from view. See "Adding/Removing Columns" on page 128.

Drilling Down Into Traffic Data

You can view traffic for a resource in the Site Tree, and then drill down to view related traffic for a session in the Traffic grid view.

Viewing Traffic for a Resource

You can view the traffic for a resource in the Site Tree. To view the traffic for an item:

- Select the item in the **Site Tree**.

  All traffic involving that item appears in the Traffic grid.

Viewing Related Traffic for a Session

You can view the related traffic for a session in the Traffic grid.

To view related traffic for a session:

- Double-click a session in the **Traffic** grid.

  The Related Traffic grid appears. If parent traffic sessions are available, you can click through the list of parents and see the HTTP and browser detail views for them.

To return to the Traffic grid:

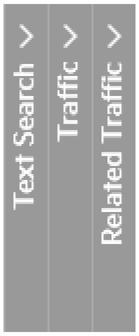
- Click the vertical **Traffic** title bar.

  The Traffic grid appears displaying all traffic.

  For more information, see "Working with Stacked Grids" below.

Working with Stacked Grids

When you drill down into grid data, an additional grid opens with a vertical title bar. When you drill down through multiple layers of grid data, each new grid is stacked on the previous grid with its vertical title bar visible. The following example shows three stacked grids.



Note: Not all applications include all of the grids shown above.

Viewing and Closing Stacked Grids

You can view a specific grid in the stack by closing any grids stacked on it. You can also close all stacked grids at once.

To view a specific grid in the stack:

- Click the title bar of the grid you want to view.

  All grids stacked on the one you want to view are closed.

To close all stacked grids:

- Click the leftmost grid title bar.

  All stacked grids are closed.

See Also

"Customizing Grid Views" on page 128

Searching and Filtering

You can search on the data displayed in grid views and in most non-grid views. You can also sort and filter on each column displayed in a grid. If an active scan is being viewed, you can search, filter, and sort on live data in the scan that is running. For more information about formatting search queries, see "Understanding the Search Expressions" on page 138.

Searching in Grid Views

You can search for data in a single column or in multiple columns displayed in a grid. To search on the data displayed in a grid:

1. Click the search icon (🔍).
2. In the **Search** field, type the column name (without spaces), the operator, and the value you are searching for.

   Examples:

   ```
   Status='404 Not Found'
   ```

   ```
   ResponseStart>'9/4/2015 9:08:52.242 AM'
   Status~'3[0-9][0-9].*'
   ```
3. (Optional) To search on multiple columns, press the **Space Bar**, type the next column name (without spaces), the operator, and the value you are searching for. Searching on multiple columns is treated as an AND search; only records that include search criteria specified for each column will be displayed. Repeat for each column that you want to search.

   Example:

   ```
   Method=GET Status~'3[0-9][0-9].*'
   ```
4. Press **Enter** or click 🔍.

You can also use regular expressions to search for patterns in the grid. For more information, see "Understanding the Search Expressions" on the next page.

Searching in Non-grid Views

You can search for data in non-grid views, such as in the Request and Response tabs. To search in tabs:

1. Select a row of data in the grid.

   Details for the selected data appear in the associated tab(s), such as in the Request and Response tabs.
2. Type the value you are searching for in the tab search field.
3. (Optional) To use regular expressions in your search criteria, select the **RegEx** check box. For more information, see "Understanding the Search Expressions" on the next page.
4. Press **Enter**.

Clearing the Search

To clear the search criteria, click the **x** in the search icon.

Sorting in the Grid

To sort by any column in the grid:

- Click the column heading.

Filtering in the Grid

To filter on one or more columns in the grid:

1. Click ▼ in the column heading.

   A filter panel appears below the column heading.
2. Type a filter expression in the filter field.

   A filter expression consists of an optional operator (`>,<,>=,<=,!=,~,=`) or one of the functions “`in`”, “`notin`”, or “`regex`” followed by a string. The range operator (`..`) is an exception, as it sits between two strings. For more information, see "Understanding the Search Expressions" on the next page.

   Examples:

   ```
   443
   '400 Bad Request'
   30*
   '9/3/2015 10:53:08.000 AM'..'9/3/2015 10:53:12.089 AM'
   in(200,300) notin(400,500)
   ```

   > **Note:** The equal (=) operator may not filter accurately on columns containing date and time information.

   For more information, see "Rules for Filtering in the Grid" on the next page.

3. Press **Enter**.

   Data in the grid is filtered based on the expression entered. The icon in the filtered column heading changes to ▼.

4. To filter on additional columns, repeat steps 1-3 on each column.

Rules for Filtering in the Grid

The following rules apply to filtering in the grid:

- You do not need to specify the field name. Since you edit the filter in a specific column, the field name is identified implicitly.
- You can use search operators in the filter field. For more information, see "The Operators" on page 140.
- If no operators or wild cards are specified in the filter field, the filter is converted to a "contains" clause of the form `field:*string*`. If the search is enclosed in quotation marks, the filter is converted to `field:'*string*'`.

  > For example, the filter string `404 Not Found` in the Status column is converted to `Status:'*404*' Status:'*Not*' Status:'*Found*'` and displays all sessions with a Status that contains either 404, Not, or Found. The filtered results would include such statuses as '302 Found', '404 Not Found', and '405 Method Not Allowed'.
  >
  > The filter string `'404 Not Found'` in the Status column is converted to `Status:'*404 Not Found*'` and displays all sessions with a Status that contains '404 Not Found'.

- You can specify multiple search filters in the filter field, separated by spaces.
- Filters on date and time fields must be enclosed in either single (') or double (") quotation marks.

Clearing a Filtered View

To clear a filtered view on one or more columns in the grid:

1. Click ▼ in a column heading that is filtered.

   A search panel appears below the column heading.

2. Click **Clear**.

   Data in the column is no longer filtered.

3. To clear the filter on additional columns, repeat steps 1 and 2 on each filtered column.

Understanding the Search Expressions

This topic explains the components of the expressions used to search in the grid and tabs.

Basic Format of a Query

The basic format of a search query is:

<PropertyName><Operator><SearchValue>

If you are searching the entire grid, the PropertyName is the column name that you wish to include in the search. If you are searching in a tab, such as the Request or Response tabs, the PropertyName is the field/property name, such as 'Request' or 'Response'.

If you are searching within a column in the grid, omit the PropertyName. The format for this type of search is:

<Operator><SearchValue>

To use regular expression (RegExp) syntax in your search, the format is:

<PropertyName> RegExp('[*RegexSearchValue*]','[*RegexFlags*]')

For more information about using regular expressions, see "Using Regular Expressions" on page 142.

Simple Query

You can perform a simple query on string data that contains no special characters and on integers. Simple queries are:

```
Method=GET

Scan.CheckId=6
```

Show me on YouTube.

Searching for Data that Contains Spaces or Special Characters

If there is a space or special character in the content you are searching for, enclose the content in either single (') or double (") quotation marks:

```
 Status='404 Not Found'

Path='/signin.html'
```

Show me on YouTube.

The quotation marks can be combined with wildcards:

```
ResponseStart:*'7/8/2015 4:22:'*
```

Searching with More than One Expression

A search can include more than one expression at the same time, with the expressions separated from each other by a space:

```
Path='/banklogin.asp' Method=GET
```

Show me on YouTube.

If the same field is listed more than once, it becomes an "OR" expression:

```
Path='/banklogin.asp' Path='/login1.asp'
```

This search would return all records where Path is either '/banklogin.asp' or '/login1.asp'.

Other fields added to the expressions are treated as an "AND" expression:

```
Path='/banklogin.asp' Path='/login1.asp' Method=POST
```

This search would return all records where Path is either '/banklogin.asp' or '/login1.asp' AND Method is 'POST'.

Another example of an AND/OR search is:

```
Method=POST Scan.Engine:Sql* Scan.Engine:Cross*
```

This search would return all records where Method is 'POST' and the value of Scan.Engine starts with either 'Sql' or 'Cross'.

Show me on YouTube.

Searching for Null Data

To search for data that contains null (empty) entries, use the = operator followed by two single quotation marks (''):

```
ParameterValue=''
```

To filter for data that contains null (empty) entries in a specific column, use the = operator followed by two single quotation marks ('') in the column filter field.

Show me on YouTube.

Using Column Names in Search Queries

To search on a column or field name that includes a space, remove the space in the search query. For example, to search on the Response End column in the grid, use the following format:

```
ResponseEnd='7/8/2015 4:22:52 PM'
```

Using Regular Expressions

To search for patterns, you can use the regular expression operator (~) and include regular expressions in the search:

```
Response~'[0-9].*='
```

Show me on YouTube.

You can also construct regular expression syntax:

```
Response RegExp('[0-9].*=','i')
```

For more information about using regular expressions, see "Using Regular Expressions" on page 142.

The Operators

The following table describes the operators and functions available for use in searching and filtering. The PropertyName used in the example column would be the column name when searching the grid or the field/property name when searching tabs. If you are filtering directly in a column, do not include the field/property name in the column filter field.

Operator	Description	Example(s)
=	Find only exact matches to the search string	`PropertyName=asdf`
>	Find data greater than the search number or date	`PropertyName>123`
>=	Find data greater than or equal to the search number or date	`PropertyName>=123`
<	Find data less than the search number or date	`PropertyName<123`
<=	Find data less than or equal to the search number or date	`PropertyName<=123`
!=	Find data not equal to the search string	`PropertyName!=asdf`
:	Find only exact matches to the search string using wildcards; search is case sensitive If the search string contains a space or dash (-), it must be enclosed in either single or double quotation marks.	`PropertyName:asdf` (find exact matches) `PropertyName:*asdf` (find data that ends with search string) `PropertyName:*asdf*` (find data that contains search string) `PropertyName:asdf*` (find data that starts with search string)
..	Find data that is within a specified range of values	`PropertyName:'7/15/2015 5:00 PM'..'7/15/2015 5:15 PM'`
~	Find the search string using regular expressions For more information about using regular expressions, see "Using Regular Expressions" on the next page.	`PropertyName~'sea[a-z]ches'`
in	Find matches to the search value(s) listed in parentheses; to search for multiple values, include a comma-separated list in parentheses	`PropertyName in(123,456)` or `PropertyName in(abc,def)` `Port in(80,443)` (find all sessions with a port of 80 or 443)

Operator	Description	Example(s)
	Show me on YouTube.	`Method in(GET)` (find all sessions with a method of 'GET')
notin	Find everything except the search value(s) listed in parentheses; to exclude multiple values, include a comma-separated list in parentheses Show me on YouTube.	`PropertyName notin(123,456)` or `PropertyName notin(abc,def)` `Port notin(80,443)` (exclude all sessions with a port of 80 or 443) `Method notin(GET)` (exclude all sessions with a method of 'GET')

Using Regular Expressions

Using the tilde (~) operator with a regular expression means that whatever is on the left of the tilde is searched using the regular expression on the right. You can also construct more complex regular expression (RegExp) syntax.

Traffic String Properties for Searching

You can use regular expressions to search any of the Traffic string properties, which are numbers, strings, or dates. This includes all fields that are listed when you click the settings icon (⚙) in a Traffic grid view.

Using the Tilde (~) Operator

When using the tilde (~) operator, the format is:

<PropertyName>~'RegexPattern'

You can use single or double quotation marks.

Examples

The following query returns a list of sessions with a Referer in the request header that contains an index.jsp file:

```
Request~'Referer:\\s.+/index\\.jsp'
```

The following query returns a list of sessions with a Location in the response header that contains an index.php or index.html file:

```
Response~'Location:\\s.+/index\\.(php|html)'
```

The following query returns a list of sessions with index.html or index.php files that were attacked by an audit engine whose name begins with ‘Cross’ or ‘Sql’:

```
Path~'/index\.(html|php)' Scan.Engine~'^(Cross|Sql)'
```

Using RegExp Syntax

RegExp syntax, which is similar to JavaScript, uses the following formats:

<*PropertyName*> RegExp('*RegexPattern*') - Performs a case-sensitive search

<*PropertyName*> RegExp('*RegexPattern*','*i*') - Performs a case-insensitive search

Examples

The following query returns a list of sessions with a Referer in the request header that contains an index.jsp file:

```
Request RegExp('Referer:\\s.+/index\\.jsp','i')
```

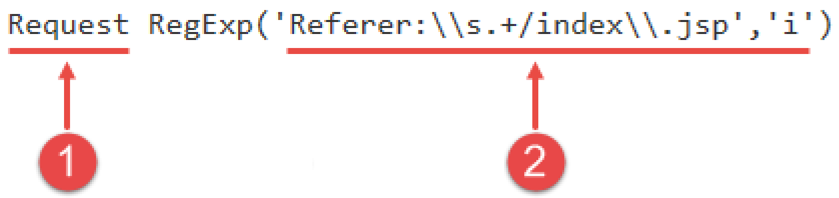
The following query returns a list of sessions with a Location in the response header that contains an index.php or index.html file:

```
Response RegExp('Location:\\s.+/index\\.(php|html)','i')
```

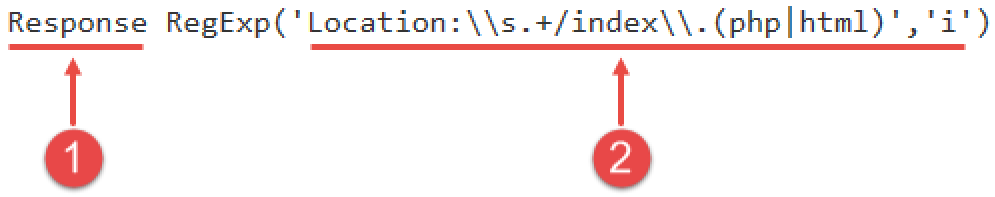
Show me on YouTube.

Understanding the RegExp Syntax

The following diagrams define the parts of the RegExp syntax.



```
Request RegExp('Referer:\\s.+/index\\.jsp','i')
```



```
Response RegExp('Location:\\s.+/index\\.(php|html)','i')
```

Item	Description
1	Specifies whether raw HTTP Request or raw HTTP Response data is searched; includes both Header and Body data
2	Defines the regular expression pattern to search for using the

Item	Description
	regular expression characters described in the table below

Regular Expressions

Special characters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used. Another recommended resource is the Regular Expression Library online at http://regexlib.com/Default.aspx.

Character	Description
\	Marks the next character as special. /n/ matches the character " n ". The sequence /\n/ matches a linefeed or newline character.
^	Matches the beginning of input or line. Also used with character classes as a negation character. For example, to exclude everything in the content directory except /content/en and /content/ca, use: /content/[^(en|ca)].*/.* . Also see \S \D \W.
$	Matches the end of input or line.
*	Matches the preceding character zero or more times. /zo*/ matches either " z " or "zoo."
+	Matches the preceding character one or more times. /zo+/ matches "zoo" but not "z."
?	Matches the preceding character zero or one time. /a?ve?/ matches the "ve" in "never."
.	Matches any single character except a newline character.
|	Indicates OR between two or more literal text search terms. For example, the following query will return a list of sessions where the path contains /index.html OR /index.php: `Path~'/index\.(html|php)'`

i	Ignores character case. Use this character in the second argument in the RegExp. For example:
	`PropertyName RegExp('stuff[abc]','i')`
	You can combine this with other flags. For example:
	`PropertyName RegExp('stuff[abc]','mi')`

m	Searches in multi-line mode. Use this character in the second argument in the RegExp. For example:

Character	Description

	`PropertyName RegExp('stuff[abc]','m')`
	You can combine this with other flags. For example:
	`PropertyName RegExp('stuff[abc]','mi')`

[xyz]	A character set. Matches any one of the enclosed characters. /[abc]/ matches the "a" in "plain."
\b	Matches a word boundary, such as a space. /ea*r\b/ matches the "er" in "never early."
\B	Matches a non-word boundary. /ea*r\B/ matches the "ear" in "never early."
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a non-digit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a linefeed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v]
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v]
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any non-word character. Equivalent to [^A-Za-z0-9_].

Using the Traffic Viewer Proxy

You can create a new proxy file using the Traffic Viewer in proxy mode. This file can be saved as a traffic file or as a macro. For example, you may want to record the login process for your website and save the captured data as a login macro.

Starting Proxy Mode

To start proxy mode, do one of the following:

- While viewing traffic data from an open scan, click **NEW**.
- After launching the Traffic Viewer from the Tools menu (or the Toolkit in Fortify WebInspect

Enterprise), click **OPEN** to view a previously recorded proxy file or click **NEW** to create a new one.

The proxy tool buttons appear at the top of the window.

Creating a New Proxy File

To create a new proxy file:

1. Click **NEW**.

   The proxy tool buttons appear at the top of the window.
2. To begin recording the proxy file, click **START**.
3. Click the **BROWSE** drop-down menu and select the browser to use.

   The tool launches the selected browser.
4. In the browser, navigate to the portions of your site that you wish to view in the proxy file.

   Traffic coming through the proxy populates the grid in the Traffic Viewer.
5. When you are finished, click **STOP**.
6. Do one of the following:
   - To save the proxy file as a traffic file (.tsf), click **SAVE**.
   - To save the proxy file as a macro (.webmacro), click the **SAVE** drop-down menu and select **as Macro**.

Configuring the Traffic Tool Proxy

The following sections describe how to configure the Traffic Tool Proxy settings.

Configuring the Proxy Listener

A proxy listener is a local HTTP proxy server that listens for incoming connections from your browser.

You configure the Proxy Listener on the settings page. Click ⚙ to access the settings.

To configure the proxy listener:

- In the GENERAL area, type the **Local IP Address** and **Port** number for the proxy listener.

Note: By default, the proxy uses localhost (IP address 127.0.0.1) and port 8080, but you can change this if necessary.

To configure Web Proxy on your host to be used by another host, you will need to change the value of the Local IP Address. The default address of 127.0.0.1 is not available to outside hosts. If you change this value to your workstation's current IP address, remote stations can use your workstation as a proxy.

Both the proxy and your Web browser must use the same IP address and port. These settings are automatically applied to the browser when you use the Browse button in proxy mode. If you launch the browser outside of the Traffic Viewer, the settings are not applied.

Configuring the Proxy

You configure the proxy settings in the application settings. Click ⚙ to access the settings.

To configure the proxy:

1. Select from the options in the **PROXY** section. The options are described in the following table.

Option	Description
Direct Connection (proxy disabled)	Select this option if you are not using a proxy server.
Auto detect proxy settings	Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.
Use System proxy settings	Import your proxy server information from the local machine.
Use Firefox proxy settings	Import your proxy server information from Firefox. **Note:** Using browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy server will not be used.
Configure proxy using a PAC file	Load proxy settings from a Proxy Automatic Configuration (PAC) file in the location you specify in the **PAC File URL** field.
Explicitly configure proxy settings	To configure a proxy, provide the following information: a. From the **Type** list, select a protocol type for handling TCP traffic through a proxy server: Socks4, Socks5, or Standard. b. If authentication is required, select one of the following types from the **Authentication Type** list: **Automatic** Allow Fortify WebInspect to determine the correct authentication type. **Note:** Automatic detection slows the scanning process. If you specify one of the other authentication methods, scanning performance will be improved. **Basic** A widely-used, industry-standard method for collecting user name and password information. i. The web browser displays a window for a user to type previously-assigned credentials. ii. The web browser then attempts to establish a connection to a

Option	Description
	server using the user's credentials. iii. If user credentials are rejected, the browser displays an authentication window to re-enter the credentials. iv. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established. The advantage of Basic authentication is that it is part of the HTTP specification and most browsers support it. The disadvantage is that web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Fortify recommends that you *not* use Basic authentication unless you are confident that the connection between the user and your Web server is secure. **Digest** The Windows Server operating system implements the Digest authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using Digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the password. In this way, an attacker cannot determine the password by sniffing network traffic. **Kerberos** Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service. **Negotiate** The Negotiate authentication protocol begins with the option to

Option	Description
	negotiate for an authentication protocol. When the client requests access to a service, the server replies with a list of authentication protocols that it can support and an authentication challenge based on the protocol that is its first choice. For example, the server might list Kerberos and NTLM, and send a Kerberos challenge. The client examines the contents of the reply and checks to determine whether it supports any of the specified protocols. If the client supports the preferred protocol, authentication proceeds. If the client does not support the preferred protocol, but does support one of the other protocols listed by the server, the client lets the server know which authentication protocol it supports, and the authentication proceeds. If the client does not support any of the listed protocols, the authentication exchange fails. **NTLM** NTLM (NT LAN Manager) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network. Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and Fortify WebInspect has to pass through a proxy server to submit its requests to the Web server, Fortify WebInspect may not be able to crawl or audit that web site. Use caution when configuring Fortify WebInspect for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem. c. In the **Server** field, type the URL or IP address of your proxy server, followed (in the **Port** field) by the port number (for example, 8080). d. If your proxy server requires authentication, type credentials in the **User Name** and **Password** fields. e. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), type the addresses or URLs in the **Bypass proxy for** field. Use commas to separate entries.

2. Click **SAVE**.

Configuring Client Certificates

Configure client certificates in the Traffic Viewer proxy settings. Click ⚙ to access the settings.

To enable client certificates and specify a certificate to use:

1. In the CLIENT CERTIFICATES area, select **Enable Client Certificates**.
2. Select the **Certificate Store** for the certificate you want to use. Options are:
   - Local Machine - The certificate store that is local to the computer and is global to all users on the computer.
   - Current User - The certificate store that is local to the current user account on the computer.

   > **Note:** Certificates used by a common access card (CAC) reader are user certificates and are stored under Current User.
3. Do one of the following:
   - To select a certificate from the "Personal" ("My") certificate store, select **My** from the drop-down list.
   - To select a trusted root certificate, select **Root** from the drop-down list.
4. Does the website use a common access card (CAC) reader?
   - If *yes*, do the following:
     1. Select a certificate that is prefixed with “(SmartCard)” from the **Certificate** list.

        Information about the selected certificate and a Pin field appear in the Certificate Information area.
     2. If a PIN is required, type the PIN for the CAC in the **Pin** field.
     3. Click **Test**.

        If you entered the correct PIN, a Success message appears.
   - If *no*, select a certificate from the **Certificate** list.

     Information about the selected certificate appears below the Certificate list.
5. Click **SAVE**.

Configuring Proxy Exclusions

You may not want certain types of files, such as image files or PDFs, to be included in the proxy data. You can exclude them from being recorded. Excluding these files allows you to focus on HTTP request/response lines and headers by removing clutter from the message body. Exclude these files in the Traffic Viewer proxy settings. Click ⚙ to access the settings.

To exclude file types:

1. In the DO NOT RECORD area, use regular expressions to type the file extension(s) that you want to exclude from capture in the proxy file.

   Example:

   .*\.jpg$,.*\.png$,.*\.bmp$

   For more information, see "Using Regular Expressions" on page 142.
2. Click **SAVE**.

Configuring Search and Replace

Search and replace allows you to create rules for locating and replacing text or values in HTTP messages coming through the proxy. This feature provides a highly flexible tool for automating your simulated attacks. Some suggested uses include:

- Masking sensitive data, such as user names and passwords
- Appending a cookie to each request
- Modifying the Accept request-header field to add or delete media types that are acceptable for the response
- Replacing a variable in the Request-URI with a cross-site scripting attack

Configure search and replace in the Traffic Viewer proxy settings. Click ⚙ to access the settings.

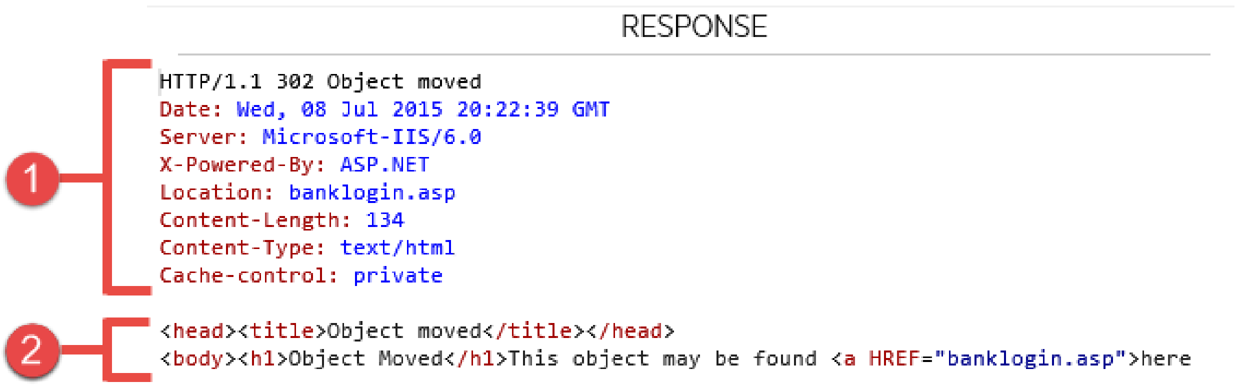
Finding and Replacing Text

To find and replace text in requests or responses:

1. Click **ADD**.

   A default entry is added to the table.
2. Double-click the **Search On** column of the entry.
3. Click the drop-down arrow and select the message area you want to search. Options are:
   - RequestFull - Search and replace in the entire request message.
   - RequestHeader - Search and replace in the request header only.
   - RequestBody - Search and replace in the request body only.
   - ResponseFull - Search and replace in the entire response message.
   - ResponseHeader - Search and replace in the response header only.
   - ResponseBody - Search and replace in the response body only.

The following diagram identifies the parts of a response message:



```
HTTP/1.1 302 Object moved
Date: Wed, 08 Jul 2015 20:22:39 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Location: banklogin.asp
Content-Length: 134
Content-Type: text/html
Cache-control: private

<head><title>Object moved</title></head>
<body><h1>Object Moved</h1>This object may be found <a HREF="banklogin.asp">here
```

Item	Description
1	Response Header
2	Response Body

4. In the **For** column, type the data (or a regular expression representing the data) you want to find.
5. In the **Replace With** column, type the data you want to substitute for the found data.

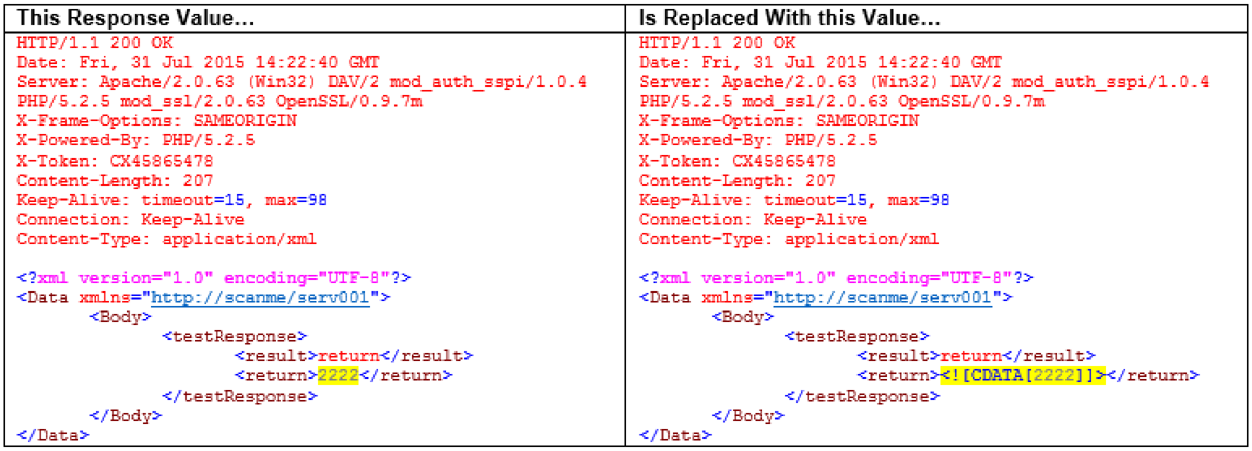
> **Note:** To use a regular expression in the **For** and/or **Replace With** columns, select the **Regex** check box. See "Using Regular Expressions in Rules" below.

6. Repeat steps 1-5 to create additional search rules.
7. Click **SAVE**.

Using Regular Expressions in Rules

> **Caution!** This section should be used only by advanced users with experience in constructing regular expression syntax.

Advanced users can configure search and replace rules using regular expressions in both the **For** column and the **Replace With** column. For example, if you enable a rule using regular expressions to search on the ResponseBody for `(<return>)([^<]+)(</return>)` and replace the findings with `$1<![CDATA[$2]]>$3`, the search rule would make the following changes:



This Response Value...	Is Replaced With this Value...
HTTP/1.1 200 OK Date: Fri, 31 Jul 2015 14:22:40 GMT Server: Apache/2.0.63 (Win32) DAV/2 mod_auth_sspi/1.0.4 PHP/5.2.5 mod_ssl/2.0.63 OpenSSL/0.9.7m X-Frame-Options: SAMEORIGIN X-Powered-By: PHP/5.2.5 X-Token: CX45865478 Content-Length: 207 Keep-Alive: timeout=15, max=98 Connection: Keep-Alive Content-Type: application/xml <?xml version="1.0" encoding="UTF-8"?> <Data xmlns="http://scanme/serv001"> <Body> <testResponse> <result>return</result> <return>2222</return> </testResponse> </Body> </Data>	HTTP/1.1 200 OK Date: Fri, 31 Jul 2015 14:22:40 GMT Server: Apache/2.0.63 (Win32) DAV/2 mod_auth_sspi/1.0.4 PHP/5.2.5 mod_ssl/2.0.63 OpenSSL/0.9.7m X-Frame-Options: SAMEORIGIN X-Powered-By: PHP/5.2.5 X-Token: CX45865478 Content-Length: 207 Keep-Alive: timeout=15, max=98 Connection: Keep-Alive Content-Type: application/xml <?xml version="1.0" encoding="UTF-8"?> <Data xmlns="http://scanme/serv001"> <Body> <testResponse> <result>return</result> <return><![CDATA[2222]]></return> </testResponse> </Body> </Data>

For more information, see "Using Regular Expressions" on page 142.

How Rules Are Applied

The request/response rules are applied sequentially, in the order in which they appear. For example, if a rule changes HTTPS to SSL, and if a subsequent rule then changes SSL to SECURE, the result will be that HTTPS is changed to SECURE.

Enabling a Rule

To enable a rule:

1. Select the **Enabled** check box for the rule you want to enable.
2. Click **SAVE**.

Disabling a Rule

To disable a rule without deleting it:

1. Clear the **Enabled** check box for the rule you want to disable.
2. Click **SAVE**.

Deleting a Rule

To delete a rule:

1. Select the rule you want to delete.
2. Click **REMOVE**.
3. Click **SAVE**.

Editing a Rule

To edit a rule:

1. Click an entry in the **Search On**, **For**, or **Replace With** column.
2. Change the data.
3. Click **SAVE**.

Chapter 15: About the Web Discovery Tool

Use Web Discovery to find all open hosts in your enterprise environment.

How It Works

Web Discovery sends packets to all the open ports (in a range of IP addresses and ports that you specify), searches the server's response for specific information, and then displays the results. There are two predefined packets included with Web Discovery: Web Server and SSL Web Server. They both contain the following HTTP request:

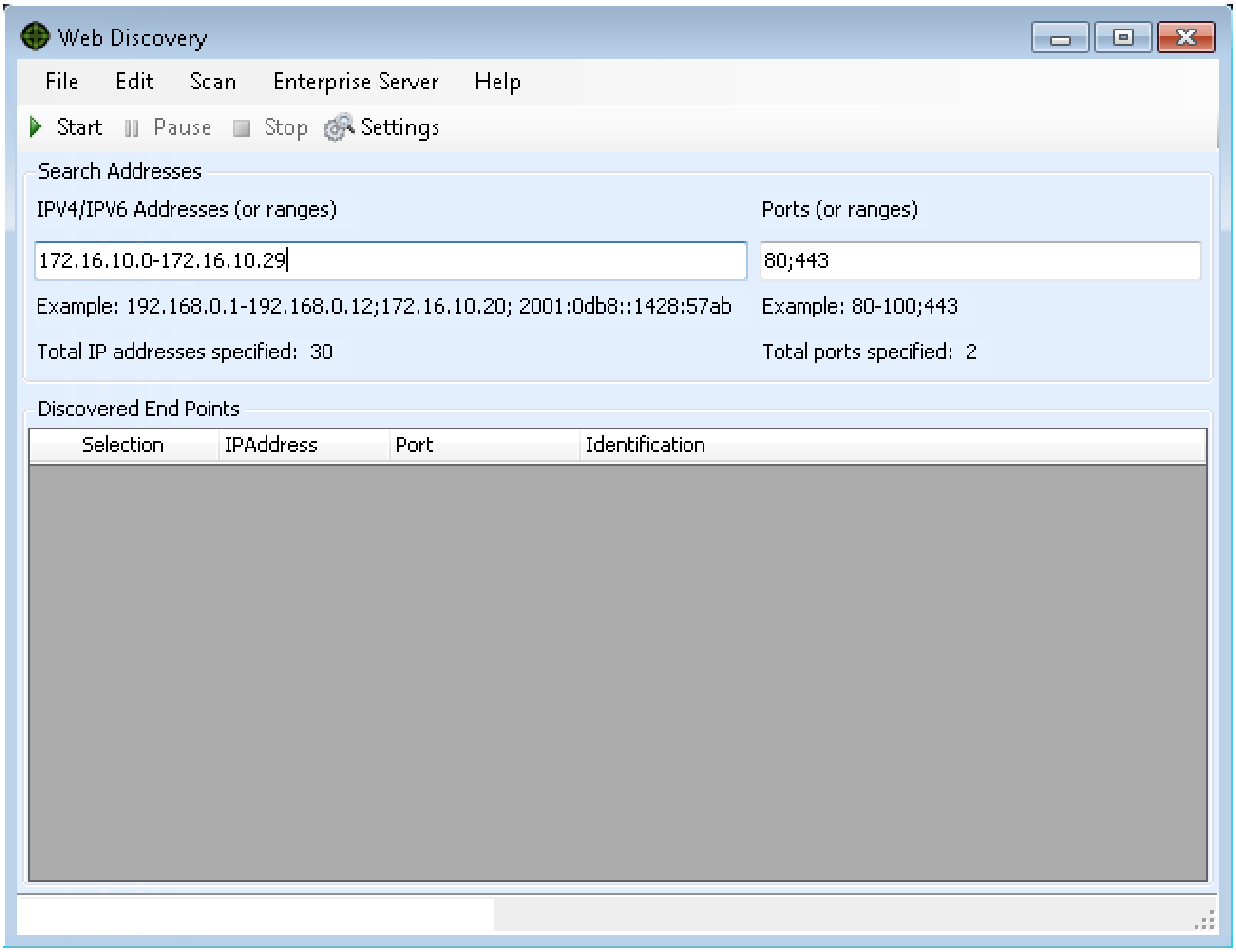
```
GET / HTTP/1.0
```

Web Discovery searches the HTTP response for the string "HTTP"; if it finds the string, it displays the IP address, port number, and the text "WebServer," followed by the results of a regular expression search designed to reveal the server's name and version number.

You can save the list of discovered servers in a text file.

Web Discovery Tool Image

The following image shows the Web Discovery tool with an IP address range entered for discovering sites.



Search Addresses

IPV4/IPV6 Addresses (or ranges): 172.16.10.0-172.16.10.29

Example: 192.168.0.1-192.168.0.12;172.16.10.20; 2001:0db8::1428:57ab

Total IP addresses specified: 30

Ports (or ranges): 80;443

Example: 80-100;443

Total ports specified: 2

Discovered End Points

Selection	IPAddress	Port	Identification

Discovering Sites

To run Web Discovery to discover sites:

1. In the **IPV4/IPV6 Addresses (or ranges)** box, type one or more IP addresses (or a range of IP addresses).
   - Use a semicolon to separate multiple addresses.

     Example: 172.16.10.3;172.16.10.44;188.23.102.5
   - Use a dash or hyphen to separate the starting and ending IP addresses in a range.

     Example: 10.2.1.70-10.2.1.90

   Note: IPV6 addresses must be enclosed in brackets. For example:

- For http://[::1]

  Fortify WebInspect scans "localhost."

- For http://[fe80::20c:29ff:fe32:bae1]/subfolder/

  Fortify WebInspect scans the host at the specified address starting in the "subfolder" directory.

- For http://[fe80::20c:29ff:fe32:bae1]:8080/subfolder/

  Fortify WebInspect scans a server running on port 8080 starting in "subfolder."

2. In the **Ports (or ranges)** box, type the ports you want to scan.
   - Use a semicolon to separate multiple ports.

     Example: 80;8080;443
   - Use a dash or hyphen to separate the starting and ending ports in a range.

     Example: 80-8080.
3. To modify Web Discovery settings, click **Settings**. See "Settings" below for more information.
4. Click **Start** to initiate the discovery process.

   Results display in the Discovered EndPoints area.
5. Click an entry in the **IP Address** column to view that site in a browser.
6. Click an entry in the **Identification** column to open the Session Properties window and view the raw request and response.

Saving Discovered Sites

To save the list of discovered servers:

1. Click **File > Export**.

   If you export the data to a .csv file, the IP addresses become default Fortify Software Security Center applications. You can edit those applications and their associated data in Excel. In Fortify WebInspect Enterprise, you can then import the applications into Fortify Software Security Center. For more information, see the Fortify WebInspect Enterprise online Help.
2. Use the standard file-selection window to name and save the file.

See Also

"Settings" below

Settings

To change the Web Discovery tool settings:

1. Click **Edit > Settings**.
2. In the **Select Protocols** group, choose the packets you want to send by selecting or clearing the

check box next to the protocol name.

3. In the **Logging** group, select the elements you want to log:
   - **Log Open Ports**: Logs all available ports found open on the host; saves only Web server information in log file.
   - **Log Services**: Logs all services identified during the discovery.
   - **Log Web Servers**: Logs Web servers identified.
4. Enter the file location in the **Log To** box, or click the ellipsis button and use the standard file-selection window to specify the file in which the log entries should be recorded.
5. In the **Connectivity** group, set the following timeouts (in milliseconds):
   - **Connection Time Out**: The period of time that Web Discovery will wait before stopping a port scan when no information has been returned from an IP address.
   - **Send Time Out**: When sending a message to the remote IP endpoint[1], the transmission is divided into smaller packets. If the IP endpoint does not acknowledge receipt of a sent packet within the specified period of time, the socket is closed and the discovery for that endpoint reports no services.
   - **Receive Time Out**: When sending a message to the remote IP endpoint, the transmission is divided into smaller packets. If the Web Discovery tool does not receive the sent packet within the specified period of time, the socket is closed and the discovery for that endpoint reports no services.
6. Adjust the number of open sockets using the **Sockets** box. A higher number of open sockets results in a faster scan. However, a setting that exceeds a server's threshold may result in false positives.
7. Click **OK** to save the updated information and return to the Web Discovery window.

See Also

"About the Web Discovery Tool" on page 156

[1] (The name for the entity on one end of a transport layer connection; the point at which a service connects to the network. In a service-oriented architecture, any single network interaction involves two endpoints: one to provide a service and the other to consume it. In Web services, an endpoint is specified by a URI.)

Chapter 16: About the Web Form Editor

Most Web applications contain forms composed of input controls (text boxes, buttons, drop-down lists, etc.). Users generally "complete" a form by modifying its input controls (such as entering text or checking boxes) before submitting the form to an agent for processing. Usually, this processing will lead the user to another page or section of the application. For example, after completing a logon form, the user will proceed to the application's beginning page.

For the scanner to navigate through all possible links in the application, it must be able to submit appropriate data for each form.

With the Web Form Editor, you can create or modify a file containing the names of all input controls and the associated values that need to be submitted during a scan of your Web site. These entries are categorized by URL, so even if different controls on different pages have the same name, the Web Form Editor can discriminate between them. Alternatively, you can designate a form entry as "global," meaning that its value will be submitted for any input control having the same name attribute, regardless of the URL at which it occurs.

During a scan, if the scanner encounters an input control whose name attribute is not matched in the file you create, it will submit a default value (12345).

There are two ways to create a list of form values:

- Create the list manually.
- Record the values as you navigate through the application.

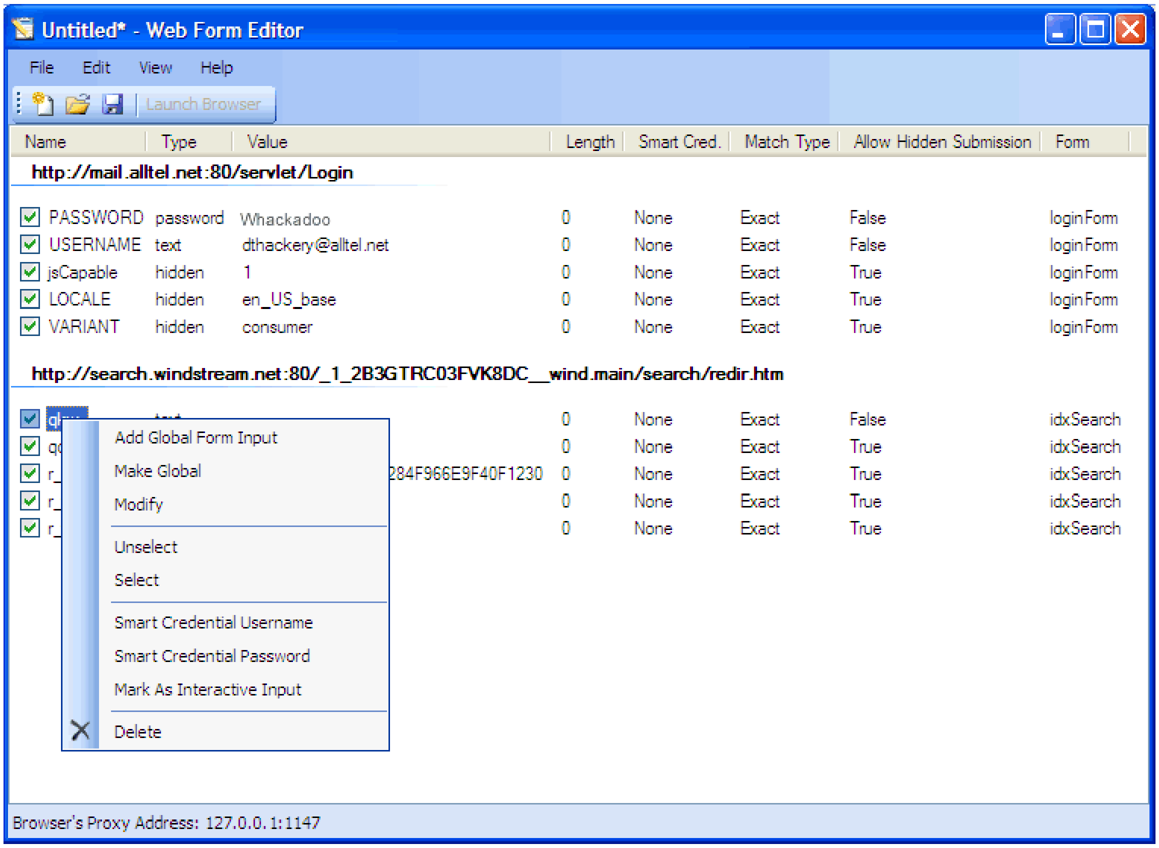
Record Web Form Values

The Web Form Editor serves as a proxy that handles HTTP traffic between a browser and a target Web site. By default, it uses the local IP address 127.0.0.1 and any available port. However, you can specify a different IP address and port by selecting **Settings** from the **Edit** menu.

Use the following procedure to capture names and values of input controls on a Web site.

1. To create a list of form values, select **New** from the **File** menu (or click the New icon on the toolbar).
2. To add form values to an existing list, select **Open** from the **File** menu (or click the Open icon on the toolbar) and choose a file using the standard file-selection dialog box.
3. Using the browser's Address bar, enter or select a URL and navigate to a page containing a form.
4. Complete the form and submit it (usually by clicking a button such as **Log In**, **Submit**, **Go**, etc.).

5. Navigate to additional pages and submit forms until you have traversed all the links you wish to follow.


Untitled* - Web Form Editor
File
Edit
View
Help
Launch Browser
Add Global Form Input
Make Global
Modify
Unselect
Select
Smart Credential Username
Smart Credential Password
Mark As Interactive Input
Delete

Name	Type	Value	Length	Smart Cred.	Match Type	Allow Hidden Submission	Form
http://mail.alltel.net:80/servlet/Login							
☑ PASSWORD	password	Whackadoo	0	None	Exact	False	loginForm
☑ USERNAME	text	dthackery@alltel.net	0	None	Exact	False	loginForm
☑ jsCapable	hidden	1	0	None	Exact	True	loginForm
☑ LOCALE	hidden	en_US_base	0	None	Exact	True	loginForm
☑ VARIANT	hidden	consumer	0	None	Exact	True	loginForm
http://search.windstream.net:80/_1_2B3GTRC03FVK8DC__wind.main/search/redir.htm							
☑ q			0	None	Exact	False	idxSearch
☑ q			0	None	Exact	True	idxSearch
☑ r_		284F966E9F40F1230	0	None	Exact	True	idxSearch
☑ r_			0	None	Exact	True	idxSearch
☑ r_			0	None	Exact	True	idxSearch

Browser's Proxy Address: 127.0.0.1:1147

For example, the last two entries in the list illustrated above were derived from the following HTML fragment ...

```
<form name="loginForm" action="/servlet/Login" method="POST">
<input type="password" size="16" name="PASSWORD">
<input type="text" size="16" name="USERNAME" value="">
<input type="SUBMIT" value="Submit"></form>
```

... and the user entered his name and password.

6. If necessary, you can modify items by right-clicking an entry and using the shortcut (pop-up) menu.
   - To edit an entry, select **Modify**.
   - To add an entry, select **Add Global Form Input**. A Global entry is one not associated with a specific URL.
   - To remove an entry, choose **Unselect**. This removes the entry from processing, but does not delete it from the file.
   - To delete an entry, choose **Delete**.
   - To designate an entry as a smart credential, select either **Smart Credential Username** or **Smart Credential Password.** See "Smart Credentials" on page 171 for more information.

- To force scanner to pause the scan and display a window prompting the user to enter a value for this entry, select **Mark As Interactive Input**.

  When a scanner encounters an HTTP or JavaScript form, it pauses the scan and displays a window that allows you to enter values for input controls within the form, provided that the scanner's option to "Prompt For Web Form Values" is selected. However, if the scanner's option to "Only Prompt Tagged Inputs" is also selected, the scanner does not pause for user input unless a specific input control has been designated **Mark As Interactive Input** (except for passwords, which always cause the scanner to pause for input).

7. From the **File** menu, select **Save** or **Save As**.

Manually Add or Modify Web Form Values

To add or modify Web form values:

1. Do one of the following:
   - To add a Web form value, right-click anywhere in the Web Form Editor's work area and select **Add Global Form Input** from the shortcut (pop-up) menu.
   - To modify a Web form value, right-click an entry and select **Modify** from the shortcut (pop-up) menu.

   The Add User-Defined Input or the Modify Input window appears.
2. In the **Name** box, type (or modify) the name attribute of the input element.
3. In the **Length** box, enter either:
   - the value that must be specified by the size attribute, or
   - zero, for input elements that do not specify a size attribute.

   For example, to submit data for the following HTML fragment . . .

   ```
   <INPUT TYPE="password" NAME="accessID" MAXLENGTH="6">
   ```

   . . . you must create an entry consisting of accessID (Name) and specify a size of "6" (Length).
4. In the **Value** box, type the data that should be associated with the input element (for example, a password).
5. Use the **Match** list to specify how the scanner should determine if this entry qualifies to be submitted for a particular input control. The options are:
   - **Exact** - The name attribute of the input control must match exactly the name assigned to this entry.
   - **Starts with** - The name attribute of the input control must begin with the name assigned to this entry.
   - **Contains** - The name attribute of the input control must contain the name assigned to this entry.
6. Programmers sometimes use input controls with type="hidden" to store information between

client/server exchanges that would otherwise be lost due to the stateless nature of HTTP. Although the Web Form Editor will collect and display the attributes for hidden controls, the scanner will not submit values for hidden controls unless you select **Allow Hidden Submission**.

7. Click **Add** (or **Modify**).
8. If necessary, you can assign additional attributes by right-clicking an entry and using the shortcut (pop-up) menu.
    - To remove an entry, choose **Unselect**. This clears the check mark and removes the entry from processing, but does not delete it from the file.
    - To activate an entry, choose **Select**. This creates a check mark and includes the entry for processing.
    - To delete an entry, choose **Delete**.
    - To designate an entry as a smart credential, select either **Smart Credential Username** or **Smart Credential Password.** See "Smart Credentials" on page 171 for more information.
    - If you select **Mark As Interactive Input**, then the scanner will pause the scan and display a window prompting the user to enter a value for this entry (if the scan options include the settings **Prompt For Web Form Values During Scan** and **Only Prompt Tagged Inputs**).

      Note: It is not necessary to tag passwords with **Mark As Interactive Input**.

Import a File

You can import a file that was designed and created for earlier versions of Fortify WebInspect and convert it to a file that can be used by the current Web Form Editor.

1. From the **File** menu, select **Import**.

   The Convert Web Form Values window appears.
2. Click the browse button ... next to **Select File To Import**.
3. Using a standard file-selection window, locate the XML file created by an earlier version of the Web Form Editor.
4. Click the browse button ... next to **Select Target File**.
5. Using a standard file-selection window, specify a file name and location for the converted file.
6. Click **OK**.

Shortcut Menu

The following commands are available from the pop-up menu that appears when you right-click in the work area of the Web Form Editor.

Command	Description
Add Global Form Input	Displays the Add User-Defined Input window, allowing you to specify the name, length, and value of an input control. For more information, see "Manually Add or Modify Web Form Values" on page 162.
Make Global	Disassociates the selected entry from a specific URL. This means that the scanner will submit the value whenever it encounters an input control having this entry's name attribute, regardless of the control's location.
Modify	Allows you to change the name, length, value, and match type attributes of an entry.
Unselect	Clears the check box associated with an entry. The entry will not be saved and will not be added again to the list if you revisit this page on which it occurred.
Select	Enables the check box associated with an entry, assuring that the entry will be included in the saved list.
Smart Credential Username	If you designate an entry as a Smart Credential Username, the Web Form Editor will not save the value you entered. When the scanner scans the page containing the input element associated with this entry, it will substitute the user name specified in its Authentication options (or, if no user name is specified, the string "FormFillText").
Smart Credential Password	If you designate an entry as a Smart Credential Password, the Web Form Editor will not save the value you entered. When the scanner scans the page containing the input element associated with this entry, it will substitute the password specified in its Authentication options (or, if no password is specified, the string "FormFillText").
Mark As Interactive Input	For Fortify WebInspect only: Tags this entry as one requiring user input if Fortify WebInspect's options are set to **Prompt For Web Form Values During Scan** AND **Only Prompt Tagged Inputs**. When Fortify WebInspect scans the page containing the input element associated with this entry, it will pause the scan until the user enters a value for this input. This is especially useful for forms that require a unique value. Examples include an order-processing system (where a duplicate number would elicit a response such as, "That order has already been processed") and a CAPTCHA (which is a type of challenge-response test to ensure that the response is not generated by a computer).

Command	Description
Delete	Removes the selected entry from the list. The entry will not be saved; it will be added again to the list if you revisit this page on which it appeared, however.

Scanning with a Web Form File

If you designate a Web form file in the default scan settings, the scanner automatically selects that file each time you start a Web site assessment. You can override that selection, however, by choosing a different file for that specific scan.

Use the following procedure to scan a site using the list of Web form values you created.

1. Click the Fortify WebInspect **Edit** menu and select **Default Scan Settings**. The Default Settings window opens.
2. In the **Scan Settings** section, select **Method**.
3. In the **Scan Behavior** group, select **Auto-fill Web Forms During Crawl**.
4. To select a previously recorded file:
   a. Click the browse button ....
   b. Using the standard file-selection window, select a file containing the Web form value you want to use and click **Open**.
   c. (Optional) You can edit the contents by right-clicking an entry and selecting an option from the context menu.
5. To record Web form values:
   a. Click **Create New Web Form Values** .
   b. Click the **File** menu and select **New**.
   c. Click **Launch Browser**.
   d. See "Record Web Form Values" on page 160 for further instructions.
6. To edit Web form values for the selected file:
   a. Click **Edit Current Web Form Values** .
   b. See "Record Web Form Values" on page 160 for further instructions.

Matching Web Form List to Input Controls

When crawling a Web application and submitting Web form values, the Micro Focus scanner analyzes the entries in the Web form values file to determine if a value should be submitted. The logic for determining a match is represented in the following table, ordered from "most preferred" to "least preferred."

Rules for Matching Web Form Values

Values	Match Case	Description
Page-specific form values	**Exact Match** Name exact match Length exact match	The specific Web page, Web form name, and value length detected on the crawled Web page exactly match a single record in the webformvalues.xml selected for the scan.
	Partial Match Name-only match Length allows wildcard	The specific Web page and Web form name detected on the crawled Web page match a single record in the webformvalues.xml selected for the scan. The field length associated with that form value allows for submission to any field input length (wildcard field length match).
Global form values	**Exact Match** Name exact match Length exact match	The Web form name and value length detected on the crawled Web page match a single record in the Global Web form values section of the webformvalues.xml selected for the scan.
	Partial Match 1 Name exact match Length allows wildcard	The Web form name detected on the crawled Web page exactly matches a form name found in the global values section of the webformvalues.xml selected for the scan. The field length associated with that form value allows for submission to any field input length (wildcard field length match).
	Partial Match 2 Field name starts with Name value Length exact match	A Web form value in the file partially matches the field name found. All characters in the Web form value match the beginning of the Web page field name and the field length detected on the crawled Web page match the record in the Global Web form values section of the webformvalues.xml selected for the scan.
	Partial Match 3 Field name starts with Name value Length allows	A Web form value in the file partially matches the field name found. All characters in the Web form value match the beginning of the Web page field

Values	Match Case	Description
	wildcard	name and the field length for the record allows for submission to any field length (wildcard field length match).
	Partial Match 4 Name value included in field name Length exact match	A Web form value in the file partially matches the field name found. All characters in the Web form value match a portion of the Web page field name and the field length detected on the crawled Web page match the record in the Global Web form values section of the webformvalues.xml selected for the scan.
	Partial Match 5 Name value included in field name Length allows wildcard	A Web form value in the file partially matches the field name found. All characters in the Web form value match a portion of the Web page field name and the field length for the record allows for submission to any field length (wildcard field length match).
No match	Field name has no exact or partial matches to Web form values	No Web form value match was found. Submit the specified default value (Default).
No default value	The Web form values file has no default value specified	No Web form value match was made and the default value is not in the webform values file. Submit "not found."

Settings: General

Use these settings to configure which browser to use and how the browser will interact with the target web site. To access these settings, select **Edit > Settings > General**.

Setting	Description
Proxy Listener	The Web Form Editor serves as a proxy that handles HTTP traffic between a browser and a target web site. By default, it uses the local IP address 127.0.0.1 and any available port. However, you can specify a different **Local IP Address** and **Port**.

Setting	Description
	To avoid the possibility of specifying a port that is already in use, select **Automatically Assign Port**.
Advanced HTTP Parsing	Most Web pages contain information that tells the browser what character set to use. This is accomplished by using the Content-Type response header (or a META tag with an HTTP-EQUIV attribute) in the HEAD section of the HTML document. For pages that do not announce their character set, you can specify which character set the Web Form Editor should use in the **Assumed 'charset' Encoding** list.
Browser to use	Some target web sites may not work with a particular browser. To use a different browser in the Web Form Editor, select Firefox or Internet Explorer from the **Browser type** list. After clicking Record in the Web Form Editor, the selected browser opens with its default tools and menu items available for use, and without the TruClient panel. **Note:** The default browser is Firefox. After choosing a different browser, the chosen browser becomes the new default.

Settings: Proxy

Use these settings to access the Web Form Editor through a proxy server. To access these settings, select **Edit > Settings > Proxy**.

Setting	Description
Direct Connection (proxy disabled)	Select this option if you are not using a proxy server.
Auto detect proxy settings	Select this option to use the Web Proxy Autodiscovery Protocol (WPAD) protocol to locate a proxy autoconfig file and configure the browser's Web proxy settings.
Use Firefox proxy settings	Select this option to import your proxy server information from Firefox.
Use System proxy settings	Select this option to import your proxy server information from the local machine.
Configure a proxy	Select this option to load proxy settings from a Proxy Automatic

Setting	Description
using a PAC file	Configuration (PAC) file. Then specify the file location in the **URL** box.
Explicitly configure proxy	Select this option to access the Internet through a proxy server, and then enter the requested information: 1. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080). 2. Select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard. 3. If authentication is required, select a type from the **Authentication** list: **Basic** A widely used, industry-standard method for collecting user name and password information. a. The Web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials. b. The Web browser then attempts to establish a connection to a server using the user's credentials. c. If a user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. d. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established. The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that Web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your Web server is secure. **NTLM** NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network. Use NTLM authentication for servers running IIS. If NTLM

Setting	Description
	authentication is enabled, and the scanner has to pass through a proxy server to submit its requests to the Web server, the scanner may not be able to crawl or audit that Web site. Use caution when configuring a scanner for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem. **Kerberos** Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service. **Digest** The Windows Server 2003 operating system implements the Digest Authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic. **Automatic** Allow the scanner to determine the correct authentication type. **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved. **Negotiate** If both the server and client are using Windows 2000 or later, Kerberos authentication is used. Otherwise, NTLM authentication is used. This method is also known as Integrated Windows authentication. 4. If your proxy server requires authentication, enter the qualifying user name and password.

Setting	Description
	5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.
Specify Alternative Proxy for HTTPS	For proxy servers accepting HTTPS connections, select **Specify Alternative Proxy for HTTPS** and provide the requested information.

Smart Credentials

When recording web form values, you will often encounter a log-on form requiring you to enter a user name and password. You can safely use your own user name and password, provided that you designate those entries as "Smart Credentials" before saving the file. Your actual password and user name are not saved.

When scanning the page containing the input control associated with this entry, the scanner will substitute the password specified in the product's Authentication options. This would be a known user name and password that does not require security. Alternatively, if no user name or password is specified, the scanner will submit the string "FormFillText."

Chapter 17: Web Macro Recorder with Macro Engine 4.0

Fortify WebInspect and Fortify WebInspect Enterprise include one "Unified" Web Macro Recorder tool. By default, it uses event-based functionality and Firefox browser technology to record new macros. See "About Macros" below.

Guided Scan is the preferred method for scanning a site using Fortify WebInspect or Fortify WebInspect Enterprise. Fortify WebInspect also continues to provide the legacy scan now known as Basic Scan, and Fortify WebInspect Enterprise continues to provide the legacy Web Site Scan.

The Web Macro Recorder can be launched in several ways—while configuring a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise, or outside of any scan in what is known as "stand-alone" mode. For more information, see "Accessing the Web Macro Recorder" on page 174.

Macros that were *recorded* in a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise can be *used* in any of those scans.

About Macros

A login macro is a recording of the events that occur when you access and log in to a website. You can subsequently instruct the Fortify scanner to begin a scan using this recording. A workflow macro is a recording of login steps (as needed) and specific URLs on a site.

> **Note:** The term "scanner" is often used instead of "Fortify WebInspect and Fortify WebInspect Enterprise" where the information applies to both products.

Firefox and IE Technology

By default the Web Macro Recorder uses underlying Firefox browser technology to record and play macros. It can also use Internet Explorer browser technology (also referred to here as IE technology) to record and display web traffic data.

Web Macro Recorder Limitations

The Web Macro Recorder does not support the recording of Flash or Silverlight applications.

The TruClient technology used in the Web Macro Recorder is an adaptation of the Ajax TruClient technology originally developed for use with Micro Focus LoadRunner and Micro Focus Performance

Center. It does not incorporate or support all the capabilities of the fully-featured version in those products.

Cookie Headers in Macros

When you play a macro, the Fortify scanner does not send any cookie headers that may have been incorporated in the recorded macro.

URLs in Macros

If a URL is in a macro, the request is always sent when the macro is played, regardless of any exclusion rules in scan settings.

McAffee SiteAdvisor May Cause Launch Error

When launching the Web Macro Recorder, you may receive the following error message:

"Exc in ev handl: TypeError: this.oRoot.enable is not a function."

This can occur if the McAfee SiteAdvisor is installed. Simply acknowledge the message and continue.

Login Macros

A login macro is a recording of the activity that is required to access and log in to a website or web application, typically by entering a user name and password and clicking a button such as Log In or Log On. When you configure a scan, you usually specify a previously recorded login macro or record a new one at the time for the scan to use.

To prevent the scanner from terminating prematurely if it gets logged out of your application, a login macro should also specify at least one logout condition that definitively indicates that a logout has occurred. During a scan, the scanner can get logged out for a variety of reasons, including:

- Normal logout driven by the target site
- An error condition in the target site such as a timeout
- An error in the macro itself, such as an invalid parameter

Specifying a logout condition as part of the login macro makes it unnecessary for users to manually log back in, perhaps repeatedly, when unexpected logouts occur during a scan. When scanning a site, the scanner analyzes every target site response to determine the state. If the scanner determines at any time that it is logged out, it runs the login macro to log back in, and then it resumes crawling or auditing the site at the point where the logout occurred.

As the final step in recording a login macro, the Unified Web Macro Recorder uses sophisticated analysis to try to *automatically* detect a logout condition and specify it in the login macro. In most cases you do not have to identify a logout condition manually. However, you can add or edit logout conditions.

Additionally, for Guided Scan login macros, to prevent potential timing issues in macro playback, you can specify that a particular object must appear in the initial Web page that appears immediately after login.

You can specify multiple logout conditions, and if any of them are met, Fortify WebInspect plays the login macro to log the scanner back in and resume the scan where it left off.

Workflow Macros

A workflow macro is a recording of the login steps (as needed) and the specific URLs to which you manually navigate on a site. When you configure a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise, you specify a previously recorded workflow macro or record a new one at the time for the scan to use. Fortify WebInspect or Fortify WebInspect Enterprise audits only the URLs that are recorded in the workflow macro and does not take any hyperlinks encountered during the audit. This type of macro is used most often to focus on a particular subsection of an application. In terms of the macro recording process, the essential differences from login macros are that:

- Workflow macros include only the specific URLs to which a user navigated while recording them. Workflow macros access only those URLs upon replay.
- Workflow macros do not require logout conditions, so the macro recorder user interface excludes logout condition functionality when recording workflow macros.

Note: If your website requires authentication, do not record login steps in a workflow macro. Instead, record a separate login macro to log in to your website.

Accessing the Web Macro Recorder

The Web Macro Recorder can be launched in several ways—while configuring a Guided Scan, a Basic Scan in Fortify WebInspect, a Web Site Scan in Fortify WebInspect Enterprise, or outside of a scan in "stand-alone" mode. For login macros and workflow macros, the following sections describe how you can record a new macro or select (and optionally edit) an existing macro that was recorded in Fortify WebInspect or Fortify WebInspect Enterprise.

Login Macros

You can record a new login macro or select (and optionally edit) an existing login macro that was recorded in Fortify WebInspect or Fortify WebInspect Enterprise in the following ways:

- When configuring a Guided Scan, specify that the target site requires a login macro, and click **Create** to record a new login macro or select (and optionally edit) an existing login macro.
- When configuring a Basic Scan in Fortify WebInspect or a Web Site Scan in Fortify WebInspect Enterprise, in Step 2 select **Site Authentication** and record a new login macro or select (and optionally edit) an existing login macro.
- In Fortify WebInspect default scan settings (**Edit > Default Scan Settings**), click **Authentication**,

select **Use a login macro for forms authentication**, and record a new login macro or select (and optionally edit) an existing login macro.

- On the Fortify WebInspect toolbar, click **Tools > Login Macro Recorder** to run the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.
- In the Fortify WebInspect Enterprise Scan Wizard, click **Advanced Settings**, click **Method** under SCAN SETTINGS, and select **Use a login macro for forms authentication** (for existing macros only).
- In Fortify WebInspect Enterprise, on the Administrative Console toolbar, click **Tools > Login Macro Recorder** to open the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.
- Using the Security Toolkit, click **Start > All Programs > Fortify > Fortify Security Toolkit > Login Macro Recorder** to run the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.
- From Windows Explorer, navigate to a particular recorded login macro and double-click to open it in the Web Macro Recorder in stand-alone mode, and record a new login macro or open (and optionally edit) an existing login macro.

Workflow Macros

You can record a new workflow macro or select (and optionally edit) an existing workflow macro that was recorded in Fortify WebInspect or Fortify WebInspect Enterprise in the following ways:

- When configuring a Guided Scan, specify that the **Scan Type** is **Workflows** and later, in the **Workflows > 1. Manage Workflows** step, record a new workflow macro or import (and optionally edit) an existing workflow macro.
- When configuring a Basic Scan in Fortify WebInspect, in Step 1 select **Workflow-Driven Scan** and click **Record** or **Manage** to record a new workflow macro or select (and optionally edit) an existing workflow macro.
- When configuring a Web Site Scan in Fortify WebInspect Enterprise, in Step 1 select **Workflow-Driven Scan** and click **Import** or **Manage** to select an existing workflow macro.
- In Fortify WebInspect Enterprise, on the Administrative Console toolbar, click **Tools > Workflow Macro Recorder** to open the Web Macro Recorder in stand-alone mode, and record a new workflow macro or open (and optionally edit) an existing workflow macro.
- Using the Security Toolkit, click **Start > All Programs > Fortify > Fortify Security Toolkit > Workflow Macro Recorder** to run the Web Macro Recorder in stand-alone mode, and record a new workflow macro or open (and optionally edit) an existing workflow macro.

About the Web Macro Recorder Interface

This topic describes the Web Macro Recorder user interface and technologies available for use when recording or editing a macro.

Firefox and IE Technologies

Fortify strongly recommends initially using the default Firefox technology of the Web Macro Recorder to record a macro. However, if Firefox has not worked for you to successfully record a macro, you can try using IE technology, which displays web traffic data in the Web Macro Recorder interface as you record and play the macro. To record a macro using IE technology, go to "Using Internet Explorer Browser Technology" on page 192 and return to this procedure when instructed to do so.

Step-by-Step Guidance

In the Web Macro Recorder, step-by-step guidance is provided near the top of the screen in a yellow instruction bar.

Toolbar

When you begin the specific process to record or edit a macro for Guided Scan, a Basic Scan in Fortify WebInspect, a Web Site Scan in Fortify WebInspect Enterprise, or stand-alone Web Macro Recorder operation, a toolbar appears with the following buttons in the Record/Edit Login Macro group for login macros or the Record/Edit Workflow group for workflow macros, except as noted:

Button/Functionality	Guided Scan	Basic Scan / Web Site Scan	Tools Menu (Standalone Mode)
New. Starts creating a new macro.	✓	✓	✓
Import. Allows you to select an existing macro to play and edit.	✓		
Open. Allows you to select an existing macro to play and edit.		✓	✓
Export. Saves the current macro under the same name or a new name.	✓		
Save. Saves the current macro under the same name or a new name.		✓	✓
Parameters Editor. See "Parameters Editor" on page 200. **Note:** This option is not available for a macro that uses IE technology in the Web Macro Recorder or	✓	✓	

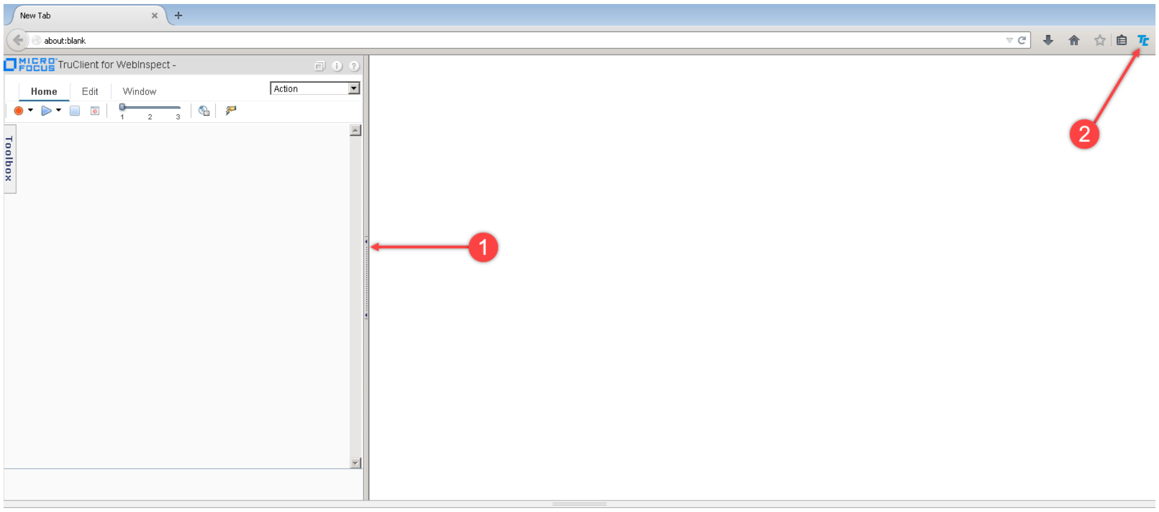
Button/Functionality	Guided Scan	Basic Scan / Web Site Scan	Tools Menu (Standalone Mode)
for a macro that was recorded using the Traffic-Mode Web Macro Recorder tool from earlier versions of Fortify WebInspect, Fortify WebInspect Enterprise, or AMP.			
Logout Conditions. See "Logout Condition Editor" on page 190. (For login macros only.)	✓	✓	✓
Browser Settings. See "Browser Settings" on page 199.		✓	✓
Rendering engine: Firefox or **Rendering engine: IE**. See "Using Internet Explorer Browser Technology" on page 192.	✓	✓	✓

Support for Tabs

The Web Macro Recorder supports the use of tabs. If a Web application uses tabs rather than launching additional browser windows, the tabs appear in the Web Macro Recorder as you record your steps through the application. When you playback the macro, the tabs open as they did when you recorded it.

Show/Hide TruClient Sidebar

If you have a Web application that needs to use the entire browser window when recording or playing back a macro, you may want to hide the TruClient sidebar. The TC button works as a toggle to hide and display the TruClient sidebar. The following image identifies where the TruClient sidebar (1) and TC button (2) are located in the Web Macro Recorder.



Recorded Locations

At times while you record or play back a macro in Guided Scan, at the bottom of the Guided Scan screen you can click the **Recorded Locations** button to expand (or later contract) the list of locations the macro has accessed. The list has the following columns:

- **Run**. Steps that are selected (checked) are played when you click **Play All**. All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
- **Excluded**. Select **Url, Directory,** or **Page** to add that type of exclusion rule. The exclusion rule will apply to any requests made by a scan that uses this scan configuration. This column also displays, read only, the causes of any existing exclusions for requests—Custom, Disallowed Host, or, if **Restrict to folder** was selected at the start of configuring the scan, Outside Root.
- **Method**. The method of the request, for example, GET or POST.
- **Status**. The status code of the response to the request, for example, 302 or 200.
- **URL**. The URL of the request.

Note: If your website requires authentication, do not record login steps in a workflow macro. Instead, record a separate login macro to log in to your website.

How to Record or Edit a Macro

This topic describes the tasks involved in interactively recording or editing login macros and workflow macros, using the Web Macro Recorder in stand-alone mode or as it is invoked when recording or editing a macro during configuration of a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise.

For information about accessing the Web Macro Recorder under a variety of circumstances, see "Accessing the Web Macro Recorder" on page 174. For information about the user interface and technologies available for use when recording or editing a macro, see "About the Web Macro Recorder Interface" on page 175.

Differences Between Login Macros and Workflow Macros

Some instructions differ between login macros and workflow macros. For example, in workflow macros, you choose where to navigate through the target site, and the URLs are recorded as you navigate, so that you can later replay them by running the macro. Also, workflow macros do not include any logout conditions.

Workflow Macros in a Guided Scan

In a Guided Scan, if you record or edit a workflow macro for a site that already has a login macro that you specify (in **Application Authentication > 1. Select Login Macro**), the login macro automatically plays before you begin recording the workflow macro to ensure that the site is accessible and to obtain and record state information from the site that the workflow macro uses whenever it is played.

Guided Scan in Fortify WebInspect Enterprise

In Fortify WebInspect Enterprise, the first time a user launches Guided Scan (or creates a report) from Fortify WebInspect Enterprise or Fortify Software Security Center, the Fortify WebInspect Enterprise Thin Client application, including an installation wizard and its own Help system, is automatically downloaded and installed on the user's computer. Then the interface for the selected function opens and its Help becomes available.

Process Overview

The process for recording or editing a macro involves the following tasks:

1. Begin based on the activity you want to perform, as follows:
   - Record/Edit a Login Macro Using Web Macro Recorder in Standalone Mode.
   - Record/Edit a Workflow Macro Using Web Macro Recorder in Standalone Mode.
   - Create/Use a Login Macro in a Guided Scan.
   - Create/Use a Workflow Macro in a Guided Scan.

- Create/Use a Login Macro in a Basic Scan or Web Site Scan.
- Create/Use a Workflow Macro in a Basic Scan or Web Site Scan.

2. Play the macro.
3. Identify a “logout” condition if automatic detection by the macro recorder fails.
4. (Optional) Modify the logout conditions.
5. (Optional) Parameterize the macro.
6. (Optional) Save the macro.

Record/Edit a Login Macro Using Web Macro Recorder in Standalone Mode

To use the Web Macro Recorder in stand-alone mode (not in conjunction with running a scan) to record or edit a login macro:

1. On the Fortify WebInspect or Fortify WebInspect Enterprise toolbar, click **Tools > Login Macro Recorder**.
2. Follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.
3. When you are instructed to play the macro, go to "Play the Macro " on page 183.

Record/Edit a Workflow Macro Using Web Macro Recorder in Standalone Mode

To use the Web Macro Recorder in stand-alone mode (not in conjunction with running a scan) to record or edit a workflow macro:

1. On the Fortify WebInspect or Fortify WebInspect Enterprise toolbar, click **Tools > Workflow Macro Recorder**.
2. Follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.
3. When you are instructed to play the macro, go to "Play the Macro " on page 183.

Create/Use a Login Macro in a Guided Scan

To create or use a login macro when completing the **Application Authentication > 1. Select Login Macro** step of a Guided Scan:

1. Select the **Use a login macro for this site** option, and do one of the following:
   - To record a new login macro, click **Create**.
   - To select (and optionally edit) an existing login macro to use in the scan, click the browse (**...**)

button to navigate to the macro.

> **Note:** If a particular login macro uses parameters, a table of those parameters is displayed when that login macro is selected. Edit the values of the parameters as needed.

- To clear a previously selected macro from the text box, click the **X** at the right end in the text box.

2. Proceed as follows:
   - If you are recording or editing a login macro, follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.
   - If you are selecting an existing login macro, after selecting a macro, click the **Next** button in the Guided Scan pane and continue configuring the scan.

Create/Use a Workflow Macro in a Guided Scan

If your website requires authentication, do not record login steps in a workflow macro in a Guided Scan. Instead, record a separate login macro to log in to your website.

To create or use a workflow macro when you are completing the **Start Parameters > 2. Choose Scan Type** step of a Guided Scan:

1. Select the **Workflows** option in the Scan Type section.

   > **Note:** Later in the Guided Scan, in the **Workflows > 1. Manage Workflows** step, workflow macro information is displayed in the right pane.

2. Do one of the following:
   - Click **Record** to record a new workflow macro.
   - Click **Import** to add an existing workflow macro to the list.

   > **Note:** When a macro in the list is selected, click **Edit** to edit it, **Delete** to remove it from the list, or **Export** to save it with a name and location you specify.

3. Proceed as follows:
   - If you are recording or editing a login macro, follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.
   - If you need to select one or more existing workflow macros, after adding the macros to the Workflows table, click the **Next** button in the Guided Scan pane and continue configuring the scan.

Allowed Hosts and Parameters in Workflow Macros for Guided Scan

When the first workflow macro is added to the list, its name (or default name) appears in the Workflows table and an Allowed Hosts table is added to the pane. Adding another workflow macro can add more allowed hosts. Any host that is enabled is available to all the listed workflow macros, not just the workflow macro for which it was added. The Guided Scan will play all the listed workflow macros and make requests to all the listed allowed hosts, whether or not their check boxes are selected. If a check box for an allowed host is selected, the scanner will crawl or audit the responses from that host. If a check box is not selected, the scanner will not crawl or audit the responses from that host.

In addition, if a particular workflow macro uses parameters, a table of those parameters is displayed when that workflow macro is selected in the list. Edit the values of the parameters as needed.

Create/Use a Login Macro in a Basic Scan or Web Site Scan

To create or use a login macro when you are completing Step 2 of a Basic Scan or Web Site Scan:

1. Select the **Site Authentication** option to use a login macro and do one of the following:
   - To record a new login macro, click **Record**.
   - To select (and optionally edit) an existing login macro to use in the scan, click the browse (**...**) button to navigate to the macro.

   > **Note:** If a particular login macro uses parameters, a table of those parameters is displayed when that login macro is selected. Edit the values of the parameters as needed.

2. Proceed as follows:
   - If you are recording or editing a login macro, follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.
   - If you are selecting an existing login macro, after selecting a macro complete Step 2 of the Basic Scan or Web Site Scan and continue configuring the scan.

Create/Use a Workflow Macro in a Basic Scan or Web Site Scan

To create or use a workflow macro when you are completing Step 1 of a Basic Scan or Web Site Scan:

1. Select the **Workflow-Driven Scan** option to use a workflow macro and do one of the following:
   - To record a new workflow macro, click **Record**.
   - To select (and optionally edit) an existing workflow macro to use in the scan, click **Manage**.

   > **Note:** If you click **Manage**, or after you record and save a new workflow macro, the Select Workflow-Driven Scan Macros dialog box appears. It displays workflow macro information for a list of workflow macros you select.

2. Do one of the following:
   - To add an existing workflow macro to the list, click **Import**.
   - To record a new workflow macro at this point, click **Record**.
   - To edit an existing macro, select it in the list and click **Edit**.

- To remove a macro from the list, select it and click **Remove**.
- To save a macro with a name and location you specify, select it and click **Export**.

3. Proceed as follows:
   - If you are recording or editing a login macro, follow the guidance in the yellow instruction bar to record or edit the macro. All of your actions will be recorded and displayed in the macro steps pane on the right. You can stop recording at any time.
   - If you are selecting one or more existing workflow macros, after adding the macros to the Macros table complete Step 1 of the Basic Scan or Web Site Scan and continue configuring the scan.

Allowed Hosts and Parameters in Workflow Macros for Basic Scan or Web Site Scan

When the first workflow macro is added, its name (or default name) appears in the Macros table in the dialog box and a specific entry is added to the Allowed Hosts list. Adding another workflow macro can add more allowed hosts. Any host that is enabled is available to all the listed workflow macros, not just the workflow macro for which it was added. The scanner will play all the listed workflow macros and make requests to all the listed allowed hosts, whether or not their check boxes are selected. If a check box for an allowed host is selected, the scanner will crawl or audit the responses from that host. If a check box is not selected, the scanner will not crawl or audit the responses from that host.

Parameters are not supported for workflow macros created or used in a Basic Scan or Web Site Scan. If you need to use parameters for your workflow macro, use Guided Scan.

Play the Macro

Play the macro, correcting any errors that occur during the process:

1. When you are instructed to do so, click the **Play** button in the instruction bar.

   The macro steps are highlighted as playback progresses. If the macro detects no errors (while using Firefox technology), “Replay succeeded” is displayed at the bottom of the right pane.

   If the macro had errors, see "How to Debug Macros" on page 205.
2. Answer the question “Did the macro play correctly?” In other words, indicate whether the login macro successfully logged in to the target site or the workflow macro accessed all the recorded URLs. Successful replay of the macro in step 1 does not guarantee that the macro did what you intended.

If you click Yes to indicate that the macro played correctly

Do the following:

- If you are recording a workflow macro, go to "(Optional) Parameterize the macro" on page 186. Workflows macros do not include logout conditions, so you are skipping tasks associated with logout conditions.
- If you are recording a login macro but not in a Guided Scan, the macro recorder attempts to automatically detect a logout condition. If it succeeds, the macro is complete and you can proceed to "(Optional) Modify the logout conditions" on page 186. If the macro recorder does not detect a logout condition, proceed to "Identify a “logout” condition if automatic detection by the macro recorder fails"

on page 185.

- If you are recording a login macro in a Guided Scan, the instruction bar in the macro recorder recommends that you select an object on the displayed Web page to specifically indicate successful login. Doing so adds a Wait step to the macro to wait for appearance of the object you will select. Without such a step in the macro, the macro operates on the assumption that the login fully succeeded when “document loading” of the initial website page (seen after login from the login page) completes. However, some websites transfer many updates to and from the browser (using AJAX technology, for example) before and/or after the browser renders the initial page. This can lead to timing problems in macro playback if the macro proceeds as though the initial page has been fully rendered when it has not and then the macro prematurely runs subsequent steps. To prevent these problems, it is best practice to click **Yes** and provide a consistent, positive indication of access to the initial page.
  - If you click **No** and do not specify an object that specifically indicates login succeeded, the macro recorder attempts to automatically detect a logout condition. If it succeeds, the macro is complete and you can proceed to "(Optional) Modify the logout conditions" on page 186. If the macro recorder does not detect a logout condition, proceed to "Identify a “logout” condition if automatic detection by the macro recorder fails" on the next page.
  - If you click **Yes** to specify an object that specifically indicates login succeeded, the macro recorder asks you to select that object. A Log Out button or “Welcome” text on the initial website page are common examples of objects that indicate successful access to that page. After you select the object, the macro recorder asks you to click **Play** to verify the macro again, now including the added Wait step. When the macro recorder asks you again if the macro succeeded, successful playback now means that the initial page actually displayed the object you specified.
    - If you click **Yes** to indicate that the macro with the login object played correctly, the macro recorder attempts to automatically detect a logout condition. If it succeeds, the macro is complete and you can proceed to "(Optional) Modify the logout conditions" on page 186. If the macro recorder does not detect a logout condition, proceed to "Identify a “logout” condition if automatic detection by the macro recorder fails" on the next page.
    - If you click **No** to indicate that the macro with the login object did not play correctly, continue to the following paragraphs.

If you click No to indicate that the macro did not play correctly

Do the following:

- If this is the *first* time you click **No**, the macro recorder automatically:
  - Adjusts the Script Level slider to level 3 to display and play back *all* of the macro steps. For more information, see "Modify Script Levels" on page 205.
  - Plays the macro again.
  - Asks you again whether the macro succeeded.

    Return to the beginning of step 2.
- If this is the *second* time you click **No**, the macro recorder automatically switches to IE technology (as if you started recording the macro by selecting the IE option for the **Rendering engine** button). A new pane for locations appears at the bottom of the Web Macro Recorder interface, rather than a pane for macro steps on the right. You can adjust the height of the locations pane relative to the top

(target site) pane. In a Guided Scan you can also expand and collapse the locations pane by clicking **Recorded Locations** below the pane. For more information, see "Using Internet Explorer Browser Technology" on page 192.

Identify a “logout” condition if automatic detection by the macro recorder fails

Note: For workflow macros, skip to "(Optional) Parameterize the macro" on the next page. Workflow macros do not have logout conditions.

At this point the target site is at a protected page, that is, a page that can be accessed only when a user is logged in.

To automatically resume a scan that gets logged out, a login macro needs to include at least one condition that represents getting logged out. Then when Fortify WebInspect or Fortify WebInspect Enterprise recognizes any of the logout conditions, it will automatically restart the macro to log back in and resume the scan where it left off.

After the macro plays back successfully, the Web Macro Recorder uses sophisticated analysis to try to automatically detect a logout condition. At that time, it displays “Detecting Logout Condition...” in the instruction bar. If the Web Macro Recorder succeeds in detecting a logout condition, the login macro is complete, as stated in the instruction bar, and you can proceed to "(Optional) Modify the logout conditions" on the next page.

You can view the automatically detected logout condition and add other logout conditions by clicking **Logout Conditions** in the toolbar to open the Logout Condition Editor.

Auto Redirect Logout Type

If the automatically detected logout condition is identified as the “Auto Redirect” type in the Logout Condition Editor, the Web Macro Recorder generated the displayed regular expression (regex), including the ‘Location’ header of a redirect (302), to represent the logout condition for the redirect when login state was lost.

If navigation parameters are specified in the scan settings, they are used as applicable at scan time to revise and uniquely identify the URL in the ‘Location’ header in the regex for the redirect. For information about navigation parameters, see the Fortify WebInspect help or the Fortify WebInspect Enterprise help for information about HTTP parsing in default scan settings.

If you later determine that the Auto Redirect regex does not work as well as necessary to automatically log back in to the site being scanned, you cannot edit the regex in place, but you can copy it, manually create a new Regex condition that you revise from the copy, and optionally delete the Auto Redirect regex. For more information, see "Logout Condition Editor" on page 190.

Automatic Logout Type

If the automatically detected logout condition is identified as the “Automatic” type in the Logout Condition Editor, the Web Macro Recorder detected a non-302 response, such as a 200.

If you later determine that the Automatic logout condition does not work as well as necessary to automatically log back in to the site being scanned, you can replace it by manually specifying a regular expression (Regex), an object (interface element), or a URL as a logout condition. For more information, see "Logout Condition Editor" on page 190.

Specifying a Logout Condition

If the Web Macro Recorder fails to detect a logout condition when you record a login macro, it presents an error message and offers to open the Logout Condition Editor, which is the same as clicking Logout Conditions in the toolbar. In the Logout Condition Editor, you can manually specify, in recommended order, a regular expression (Regex), an object (interface element), or a URL as a logout condition.

To specify a logout condition if the Web Macro Recorder could not detect one:

1. In the macro, navigate in the target site to a page that users consistently see when they get logged out.
2. Use the Logout Condition Editor to specify a logout condition that is unique to this page. See "Logout Condition Editor" on page 190 for more information.

(Optional) Modify the logout conditions

If you have been recording a login macro, this task is optional. It does not apply to workflow macros.

To examine or modify the logout condition, click **Logout Conditions** in the toolbar. You can specify as many different logout conditions as you need, and if any of them is met during a scan, Fortify WebInspect or Fortify WebInspect Enterprise invokes the login macro to log back in. For more information, see "Logout Condition Editor" on page 190.

(Optional) Parameterize the macro

This task is optional. To parameterize the login credentials or the URL, click **Parameters Editor** in the toolbar. For more information, see "Parameters Editor" on page 200.

(Optional) Save the macro

To save the macro for future use, click **Export** in the toolbar for a Guided Scan, or click **Save** (and then **Save** or **Save As**) in the toolbar for a Basic Scan in Fortify WebInspect or a Web Site Scan in Fortify WebInspect Enterprise. This completes the macro recording procedure.

If you are configuring a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise, close the Web Macro Recorder to return to the configuration process.

Recording a Macro for a Site with Multiple, Variable Login Questions

Challenge-response authentication is a family of protocols in which the server presents a question (the challenge) and the client must provide a valid answer (the response). In the simplest example, the challenge asks for a password and the valid response is the correct password.

Multiple Challenges

Many websites now present multiple challenges to the user. Typically, when a user first registers with a website, the site presents a list of questions to which the user provides answers that will be used for subsequent authentication. For example:

- What is your favorite color?
- What was the name of your first pet?
- In what town or city were your born?
- What was the make of your first automobile?

When the user later attempts to log in, the website presents two or more of these challenges.

Groups of Challenges

Some sites also create groups of challenges, and present questions from the groups on each new login attempt, as demonstrated in the following example.

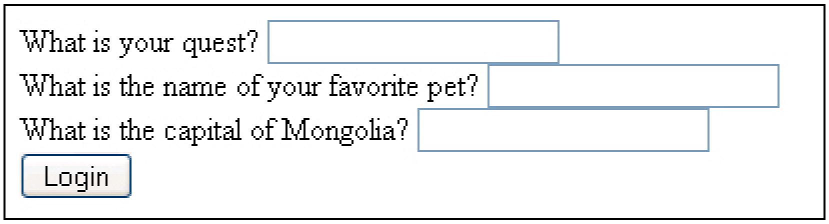
When registering for the example website, the user is asked to provide answers to nine questions, which are arranged into three groups of three questions each, as follows.

Group 1
Q: What is your quest? A: happiness
Q: What is your name? A: Smith
Q: What is your favorite color A: blue

Group 2
Q: What is the name of your favorite pet? A: Rusty
Q: What is your mother's maiden name? A: Jones
Q: In what state were you born? A: Delaware

Group 3
Q: What is the capital of Mongolia? A: Ulaanbaatar
Q: What is the name of a sea bird? A: Albatross
Q: What is your paternal grandmother's first name? A: Esther

The login page might look like this (using the first question from each group):

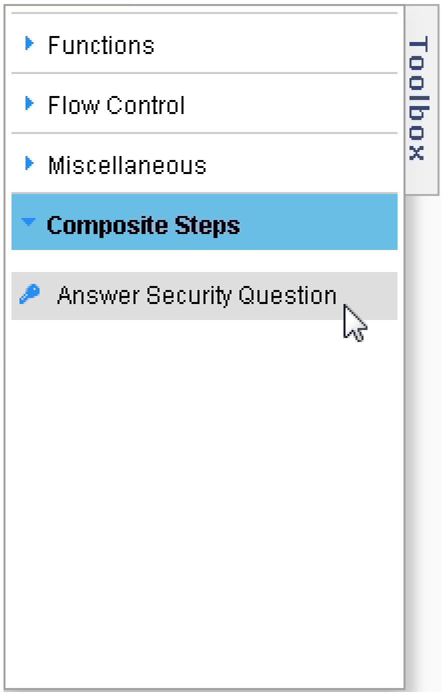

What is your quest?
What is the name of your favorite pet?
What is the capital of Mongolia?
Login

Recording a Macro for Challenge/Response Logins

When recording a macro for a challenge/response type of login, you must know all possible question-and-answer combinations, even if only a subset of those combinations might be presented during any one login. You enter these combinations manually, as special steps while recording a macro.

At the point where the target site asks the challenge questions, usually after logging in with username and password credentials, use the following procedure to manually create the required steps for this hypothetical set of nine questions:

1. If you are recording a macro, click **Stop** on the instruction bar to stop the automatic macro recording process.
2. Click the **Toolbox** vertical tab on the left side of the macro steps pane.
3. Click (expand) **Composite Steps**.

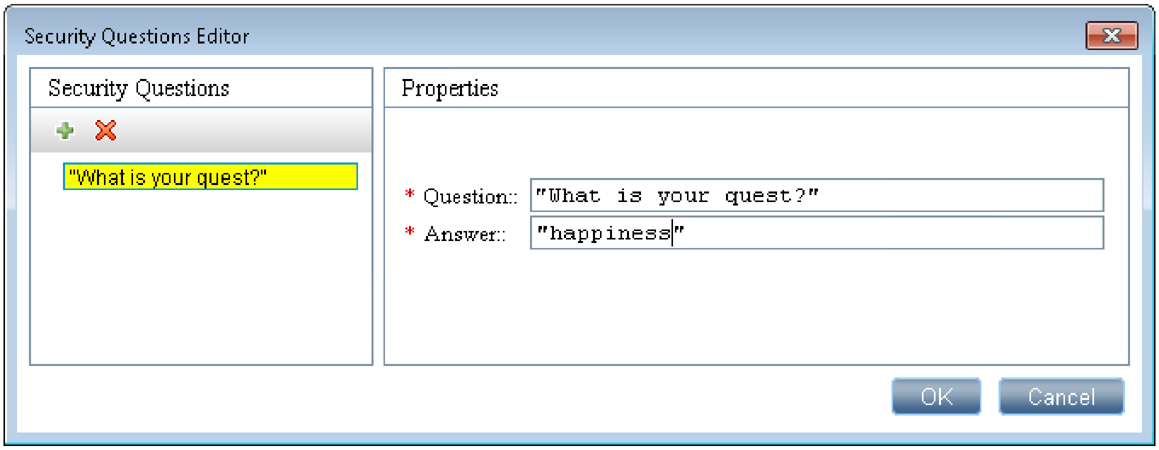


4. Click and drag the **Answer Security Questions** element to the right pane to create the next macro step.
5. Click the first “Click to choose an object” button in the new step and then, in the target site pane, click the object representing the first question (usually a label).
6. Click the second “Click to choose an object” button in the new step and then, in the target site pane, click the object representing the answer (usually a text box).
7. Place your mouse in the upper right corner of the step and click to open the Step Editor.
8. Click (expand) the **Security Questions** section.
9. Click ⊞ to open the Security Questions editor.
10. In the Security Questions pane of the Security Questions Editor, click the plus sign to add a new

question.

A new question appears with the default name "Question1." Its properties include the text box labelled **Question** (also shown with a default value of "Question1") and the text box labelled **Answer**, with a default value of "Answer1."

11. In the **Question** text box, type over the default text with the actual question exactly as it appears on the login page, including capitalization and punctuation. Be sure to enclose the text in quotation marks as shown below. The question in the left pane is simultaneously updated.



"What is your quest?"

* Question:: "What is your quest?"
* Answer:: "happiness"

12. In the **Answer** text box, enter the correct response in quotes.
13. Click **OK**.

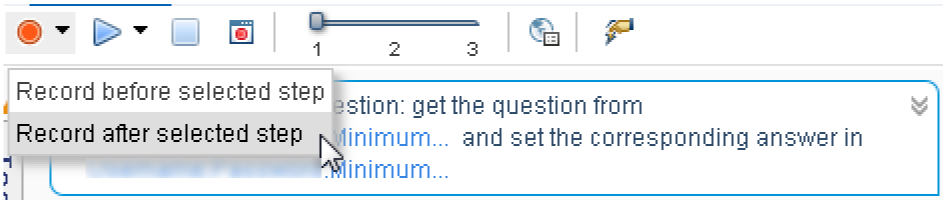
    The **Question** and **Answer** are added to a table in the **Security Questions** section in the macro step. (If you later need to edit a question or answer, reopen the Security Questions Editor.)
14. Repeat step 9 through step 13 to add the information for the second question that might appear in the same location on the web page (in this example, "What is the name of your favorite pet?").
15. Repeat step 9 through step 13 to add the information for the third question that might appear in the same location on the web page (in this example, "What is the capital of Mongolia?").

    This completes the macro step for this particular location on the web page.
16. Refresh the web page until the second set of questions appears. Click in the target site pane and press F5 (or right-click and select **Reload**).
17. Repeat step 2 through step 15 to add another macro step for the second set of three questions and answers at the second location on the web page.
18. Refresh the web page until the third set of questions appears. Click in the target site pane and press F5 (or right-click and select **Reload**).
19. Repeat step 2 through step 15 to add another macro step for the third set of three questions and answers at the second location on the web page.

20. After creating steps for all possible question-and-answer combinations, if you need to record further macro steps:

    a. Select the last step you created.

    b. Click the drop-down arrow on the **Record** button in the macro steps pane and select **Record after selected step**.

    
    1
2
3
Record before selected step
Record after selected step
estion: get the question from
Minimum... and set the corresponding answer in
Minimum...

    c. Add any further steps to the macro as needed.

    d. Click **Stop** on the instruction bar.

21. To play back the macro, return to Task 2 in "How to Record or Edit a Macro" on page 179 or, if you are using IE technology, return to Task 2 in "Using Internet Explorer Browser Technology" on page 192.

    On playback, if the macro cannot find a particular question object or answer object on the page, you can expand **Security Question Object** or **Security Answer Object** in the Step Editor for the security question and use the **Highlight** and **Replace** buttons to try to correct the failure. See "How to Resolve Object Identification Issues" on page 208.

Logout Condition Editor

The Logout Condition Editor allows you to create or edit logout conditions for login macros. For introductory information, see "Web Macro Recorder with Macro Engine 4.0" on page 172. You can specify as many different logout conditions as you need, and if any of them is met, Fortify WebInspect or Fortify WebInspect Enterprise invokes the login macro to log back in and resume a scan where it left off. The final set of all logout conditions should cover all the cases of becoming logged out during a scan of the target site.

When the Web Macro Recorder successfully detects a logout condition automatically, it categorizes the logout condition as one of the following types:

- **Auto Redirect**. This type of logout condition is created when the Web Macro Recorder detects that the target site responds with a 302 redirect. It takes the form of a regular expression (regex).
- **Automatic**. This type of logout condition is created when the Web Macro Recorder detects that the target site responds with anything other than a 302 redirect, for example, with a 200.

Adding a Logout Condition

To add a new logout condition:

1. Click the **Logout Conditions** button in the toolbar.
2. Click ⊞ in the left pane (or click the drop-down arrow to the right of ⊞ and select **Manual**).

3. In the right pane specify the name of the new condition. (Notice that the name in the left column is simultaneously updated with your changes.)
4. Select which type of logout condition you want to use and complete the information required for that type. In order of recommended priority, the options are:

Option	**Description**
Regex	With this option, you will construct a regular expression (regex). A regular expression is a pattern that describes a set of strings. Regular expressions are constructed much like mathematical expressions by using various operators to combine smaller expressions. Only users with a working knowledge of regular expressions should use this feature. The regex must reflect the difference between a) the response to a logged-in user's request to access a protected page, and b) the response to the same request from the user, while *not* logged in, to access the same protected page. The general steps to construct the regex are as follows: a. Start the Web Proxy tool to record web traffic. See the Help for that tool or the *Tools Guide for Fortify WebInspect Products*. b. Log in to the target site legitimately and copy the URL of a protected page. c. Log out and use the copied URL to try to access the protected page without logging in. d. Compare the responses and identify a unique aspect of the response to the attempt to access the protected page without logging in. e. Open the Regular Expression Editor. From the Fortify WebInspect or Fortify WebInspect Enterprise menu, select **Tools > Regular Expression Editor**. See the Help for that tool or the *Tools Guide for Fortify WebInspect Products*. f. Construct a regex that reflects the unique aspect of the response to the attempt to access the protected page without logging in. g. Copy the regex into the **Regex** field of the Logout Condition Editor.
Object	After you select this option, click **Click to choose an object**, navigate to a page where the user is logged out and can log back in, and move your mouse over objects on the page until you find one that does not appear on any other page and that indicates that the user is logged out. As you mouse over objects, each one is highlighted in green until you select one or press Esc to stop the selection process. Once you select an object, if you click **Highlight** the Logout Condition Editor is hidden temporarily and the selected object is highlighted by a rapidly flashing red outline.

Option	Description
URL	When you select this option, the currently displayed web page is automatically used as the default value. You can specify a static URL to which the target site redirects users when it logs them out. Do not specify the target site's general login page.

Detecting Logout Condition

In the Logout Condition Editor, if you click the drop-down arrow to the right of ⊞, and if the **Automatic** option is available and you select it, the Logout Condition Editor closes and the Web Macro Recorder attempts to automatically detect a logout condition.

Deleting a Logout Condition

To delete a logout condition, select it in the left pane and click the red **X**.

Using Internet Explorer Browser Technology

By default, the Unified Web Macro Recorder tries to create a macro using Firefox technology. However, if it cannot successfully create the macro, it automatically tries again using its Internet Explorer browser technology (also referred to here as IE technology), which displays locations and web traffic data in the Web Macro Recorder interface. For more information, see the procedure for recording a macro (see "How to Record or Edit a Macro" on page 179).

In the Unified Web Macro Recorder, you can also manually initiate the use of IE technology as you begin recording a new macro, in case the default Firefox technology of the Web Macro Recorder has not worked. Fortify strongly recommends that you start by trying the default Firefox technology.

The consequences of selecting IE technology (**Rendering engine: IE**) depend upon the circumstances:

- When recording a macro, selecting IE technology:
  - Switches the macro recorder from the use of custom Firefox technology to Internet Explorer browser control.
  - Records the macro in traffic mode.
- In a Guided Scan, at the **Start Parameters > 1. Verify Web Site** step or at the **Optimization Tasks > Enhance coverage of your web site** step, selecting IE technology implies that using Firefox technology would not work for any aspect of the site, so it:
  - Switches the macro recorder from the use of custom Firefox technology to Internet Explorer browser control.
  - Changes the default script execution engine so that it is compatible with Internet Explorer.

- Uses Internet Explorer as the default browser for subsequent steps in the scan such as recording a macro or discovering locations in **Active Learning**.

As you record a macro against a target site using IE technology, the Web Macro Recorder displays requested locations (as described in detail in subsequent sections). When you play a macro, the Web Macro Recorder also displays HTTP web traffic for both the requests and their associated responses, but the recorded macro includes only the requests.

Using IE Technology to Record Web Traffic

Fortify strongly recommends initially using the default Firefox technology of the Web Macro Recorder to record a macro. However, the Web Macro Recorder can invoke IE technology automatically if Firefox technology fails or, if you have not been able to successfully record a macro with the Firefox technology, you can manually invoke IE technology. Using IE technology, the bottom pane of the Web Macro Recorder interface displays requested locations. The actions you take to record and test the macro, guided by the yellow instruction bar at the top of the screen, are essentially the same as for recording a macro using Firefox technology, but the user interface is different, as described in this section.

Proceed as described in the following table, depending on how you are accessing IE technology.

If...	**Then proceed to...**
You need to initiate the use of IE technology when you are just starting to use the stand-alone Web Macro Recorder or just starting to configure a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise	"Record the Macro Using IE Technology" below
You need to edit a traffic-mode macro that was recorded in an earlier version of Fortify WebInspect or Fortify WebInspect Enterprise	"Record the Macro Using IE Technology" below
Playback failed twice using Firefox technology and the macro recorder automatically invoked IE technology	"Play the Macro Using IE Technology" on page 196

Record the Macro Using IE Technology

1. Select **Rendering engine: IE** in the Record/Edit Login Macro section of the Web Macro Recorder toolbar, the Guided Scan toolbar, the Basic Scan toolbar in Fortify WebInspect, or the Web Site Scan toolbar in Fortify WebInspect Enterprise, as applicable.

   A new pane for locations appears at the bottom of the Web Macro Recorder interface, rather than a pane for macro steps on the right. You can adjust the height of the locations pane relative to the top (target site) pane. In a Guided Scan you can also expand and collapse the locations pane by clicking **Recorded Locations** below the pane.

The buttons, check box, and columns in the locations pane are described in step 2.

2. Follow the guidance in the yellow instruction bar to record the macro.

   The instructions differ somewhat between login macros and workflow macros. For example, in workflow macros, you choose where to navigate through the target site, and the URLs are recorded as you navigate, so that you can later replay them by running the macro. Also, workflow macros do not include any logout conditions.

   Note: IE technology does not support websites that require users to answer a variable set of questions in order to log in.

   The following descriptions are provided for information and to assist with debugging. When the on-screen instructions tell you to play the macro, you can proceed to "Play the Macro Using IE Technology" on page 196.

   From the first time you navigate to a URL, a table of request data is added to the locations pane. The locations pane has a button bar with the buttons and check box described in the following table.

Button / Check Box	Description
Play Highlighted	Available after you highlight a single request (row) by clicking it. Plays the highlighted request if the associated check box in the Run column is selected. Other check boxes in the Run column do not matter. The first time you select a request, the locations pane splits into left and right panes. The left pane continues to display the table of request data for each location. In the right pane, the default Details tab is split, with HTTP request data above the associated response data.
Play All	Available after you click **Stop** in the instruction bar to stop recording the login steps. Plays only the requests that are selected (checked) in the Run column. **Note:** All steps are stored in the macro when you save it, but only the steps selected in the Run column are run whenever the macro is played.
Stop	Available during playback after you have clicked the **Play All** button. Aborts playback upon completion of the current request.
Logout	(Does not appear for workflow macros.) Available after you click **Stop** in the instruction bar to stop recording the login steps. Logs you out of the site so that you can determine how the site responds to a subsequent request you play when logged out.
Delete	Available immediately. Deletes the single request (row) you highlighted by

Button / Check Box	Description
Highlighted	clicking it.
Delete All	Available after you click **Stop** in the instruction bar to stop recording the login steps. Deletes all the requests, regardless of whether they are selected in the Run column.
Prompt for login (CAPTCHA)	(Does not appear for workflow macros.) Available immediately. CAPTCHA is a challenge-and-response test designed to ensure that a login response is provided by a person, not generated by a computer. If your target site uses CAPTCHA, select this check box. The macro still detects a logout condition, but Fortify WebInspect or Fortify WebInspect Enterprise users will need to log in manually at the beginning of a scan and whenever a logout occurs. Selecting this option disables selection of any of the listed requests and closes the right pane that displays HTTP traffic.

Below the button bar, the locations pane lists location and has the columns described in the following table.

Column	Description
Run	Steps that are selected (checked) are played when you click **Play All**. All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
Excluded	Select **Url**, **Directory**, or **Page** to add that type of exclusion rule. The exclusion rule will apply to any requests made by a scan that uses this scan configuration. This column also displays, read only, the causes of any existing exclusions for requests—Custom, Disallowed Host, or, if **Restrict to folder** was selected at the start of configuring the scan, Outside Root.
Method	The method of the request, for example, GET or POST.
Status	The status of the response to the request, for example, 302 or 200.
URL	The URL of the request.

The bottom right pane includes the tabs described in the following table.

Tab	Description
Details	For the selected (highlighted) request in the left pane, shows request data in the top right pane and associated response data in the bottom right pane.

Tab	Description
State	A collection of all the items that represent state or could represent state, that have been seen across all the locations that the macro has accessed. You can select them in any combination to characterize them as representing a state and you can manually add various types of items. Web applications can require that certain parameters be marked as “stateful.”
Parameters	(Does not appear for workflow macros.) For login macros, allows you to designate form input fields as being user name or password input so that the macro using IE technology can have user name and password parameters that can be specified at scan time, like macros that use Firefox technology.

Play the Macro Using IE Technology

Using IE technology, a pane for locations appears at the bottom of the Web Macro Recorder interface, rather than a pane for macro steps on the right. You can adjust the height of the locations pane relative to the top (target site) pane. In Guided Scan you can also expand and collapse the locations pane by clicking **Recorded Locations** at the bottom of the pane.

1. If the macro has not already been played automatically, click the **Play** button in the instruction bar.

   The following descriptions are provided for information and to assist with debugging. You can proceed to step 2.

   The locations pane has a button bar with the buttons and check box described in the following table.

Button / Check Box	Description
Play Highlighted	Plays the single request (row) you highlighted by clicking it. Plays the highlighted request if the associated check box in the Run column is selected. Other check boxes in the Run column do not matter.
Play All	Plays only the requests that are selected (checked) in the Run column. **Note:** All steps are stored in the macro when you save it, but only the steps selected in the Run column are run whenever the macro is played.
Stop	Available during playback after you have clicked the **Play All** button. Aborts playback upon completion of the current request.
Logout	(Does not appear for workflow macros.) Logs you out of the site so that you can determine how the site responds to a subsequent request you play when

Button / Check Box	Description
	logged out.
Delete Highlighted	Deletes the single request (row) you highlighted by clicking it.
Delete All	Deletes all the requests, regardless of whether they are selected in the Run column.
Prompt for login (CAPTCHA)	(Does not appear for workflow macros.) CAPTCHA is a challenge-and-response test designed to ensure that a login response is provided by a person, not generated by a computer. If your target site uses CAPTCHA, select this check box. The macro still detects a logout condition, but Fortify WebInspect or Fortify WebInspect Enterprise users will need to log in manually at the beginning of a scan and whenever a logout occurs. Selecting this option disables selection of any of the listed requests and closes the right pane that displays HTTP traffic.

Below the button bar, the left pane displays a table of data for each location and has the columns described in the following table.

Column	Description
Run	Steps that are selected (checked) are played when you click **Play All**. All steps are stored in the macro when you save it, but only the selected steps are run whenever the macro is played.
Excluded	Select **Url**, **Directory**, or **Page** to add that type of exclusion rule. The exclusion rule will apply to any requests made by a scan that uses this scan configuration. This column also displays, read only, the causes of any existing exclusions for requests—Custom, Disallowed Host, or, if **Restrict to folder** was selected at the start of configuring the scan, Outside Root.
Status	The status of the response to the request, for example, 302 or 200.
Protected	(Does not appear for workflow macros.) For login macros, a set of options with one selection allowed. The default is the request to the page that the Web Macro Recorder has identified as the most likely to be the protected page. This is also the page from which the Web Macro Recorder attempts to automatically determine a logout condition.
URL	The URL of the request.

In the right pane, the default **Details** tab is split, with HTTP request data above the associated response data for the request selected in the left pane. The right pane includes the tabs described in the following table.

Tab	Description
Details	For the selected (highlighted) request in the left pane, shows request data in the top right pane and associated response data in the bottom right pane.
State	A collection of all the items that represent state or could represent state, that have been seen across all the locations that the macro has accessed. You can select them in any combination to characterize them as representing a state and you can manually add various types of items. Web applications can require that certain parameters be marked as “stateful.”
Parameters	(Does not appear for workflow macros.) For login macros, allows you to designate form input fields as being user name or password input so that the macro using IE technology can have user name and password parameters that can be specified at scan time, like macros that use Firefox technology.

If the macro itself detects an inconsistency between an expected status code for a response as determined during macro recording and the actual status code during macro playback, the macro highlights the difference between expected and actual status in the bottom left pane. Investigate and address this condition.

2. Answer the question "Did the macro play correctly?" In other words, indicate whether the login macro successfully logged in to the target site or the workflow macro accessed all the recorded URLs. Successful replay of the macro in step 1 does not guarantee that the macro did what you intended.

   - If you click **Yes**:

     - If you are recording a *login* macro, the macro recorder attempts to automatically detect a logout condition. Return to Task 3: Identify a "logout" condition if automatic detection by the macro recorder fails, in "How to Record or Edit a Macro" on page 179, and from that task on, perform the same procedures as you would for the Web Macro Recorder using its default Firefox technology.

     - If you are recording a *workflow* macro, return to Task 5: (Optional) Parameterize the macro, in "How to Record or Edit a Macro" on page 179, and from that task on, perform the same procedures as you would for the Web Macro Recorder using its default Firefox technology. Workflow macros do not include logout conditions, so you are skipping tasks associated with logout conditions.

   - If you click **No**:

     The instructions advise you to create a new macro or use the Help. For example, see "How to Debug Macros" on page 205 or "How to Resolve Object Identification Issues" on page 208.

Browser Settings

When using the Web Macro Recorder in stand-alone mode (click **Tools > Login Macro Recorder** or **Tools > Workflow Macro Recorder** in Fortify WebInspect or in the Fortify WebInspect Enterprise Administrative Console), click the **Browser Settings** button in the toolbar to display the **Proxy Settings** and **Network Authentication** tabs, described in the following sections. For Guided Scans, Basic Scans in Fortify WebInspect, and Web Site Scans in Fortify WebInspect Enterprise, proxy settings and network authentication are configured as part of the scan.

Browser settings are not saved in macros.

Proxy Settings Tab

Select one of the following options:

Option	Description
Direct Connection (proxy disabled)	Select this option if you are not using a proxy server.
Auto detect proxy settings	Select this option to use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.
Use System proxy settings	Select this option to import the proxy server information from the local machine.
Use Firefox proxy settings	Select this option to import the proxy server information from Firefox.
Configure proxy settings using a PAC file	Select this option to load proxy settings from a Proxy Automatic Configuration (PAC) file in the location you specify in the URL box.
Explicitly configure proxy settings	Select this option to configure a proxy by entering the requested information, as follows: • **Server:** Enter the URL or IP address of your proxy server. • **Port:** Enter the port number (for example, 8080). • **Type:** Select a protocol for handling TCP traffic through a proxy server—Standard, SOCKS4, or SOCKS5. • **Authentication:** Select an authentication method. For a

Option	Description
	description of authentication methods, see the Help or the User Guide for the product. • **User Name:** Specify a user name. • **Password:** Specify a password. • **Bypass proxy for:** If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), select this option and enter the addresses or URLs in the box. Use commas to separate entries.

Network Authentication Tab

If network authentication is required:

1. Click **Network Authentication**.
2. Select one of the methods. For a description of authentication methods, see the Help or the User Guide for the product.
3. Specify a **User Name** and **Password** for network authentication.
4. Select or clear the **Client Certificate** check box. If selected, complete the Certificate Store fields and select a certificate.

Parameters Editor

When recording a macro, you can use the Parameters Editor for the following purposes:

- Creating parameters for the user name and password to allow testers to use their own authentication credentials when starting a scan or to use multiple credentials for a multi-user login scan. For procedures, see "Using Name and Password Parameters" below.
- Creating a parameter for the URL to allow testers to designate an alternate URL when the macro runs. For example, suppose you record a macro for www.testsite.com. At a later point in time, you rename the site to www.testsite2.com. If you parameterize the URL when you record the macro, you do not need to record a new macro. You simply enter a new host name as the Start URL when you run a scan. For procedures, see "Using a URL Parameter" on page 202.

Using Name and Password Parameters

Task 1: Create Parameters

1. After creating and testing your log-in macro, click **Parameters Editor** in the toolbar.

   The Parameters Editor opens.

2. Click + to add a parameter.

3. In the **Name** text box, enter a name for the parameter (for example: `Username`).
4. In the **Value** text box, enter the default text (for example: `Enter user name`) that you want testers to see as the value for the parameter as they configure a scan. However, while you are developing the macro, for macro playback to succeed you will probably need to enter your own valid user name and change it to the default text after you verify the macro near the end of this procedure.
5. Click + to add a second parameter.
6. In the **Name** text box, enter a name for the parameter (for example: `Password`).
7. In the **Value** box, enter the default text (for example: `Enter password here`) that you want testers to see as the value for the parameter as they configure a scan. However, while you are developing the macro, for macro playback to succeed you will probably need to enter your own valid password and change it to the default text after you verify the macro near the end of this procedure.
8. Select **Encrypted** if the Value should be encrypted before transmission to the web server.
9. Click **Close** to close the Parameters Editor.

Task 2: Assign Parameters to Steps

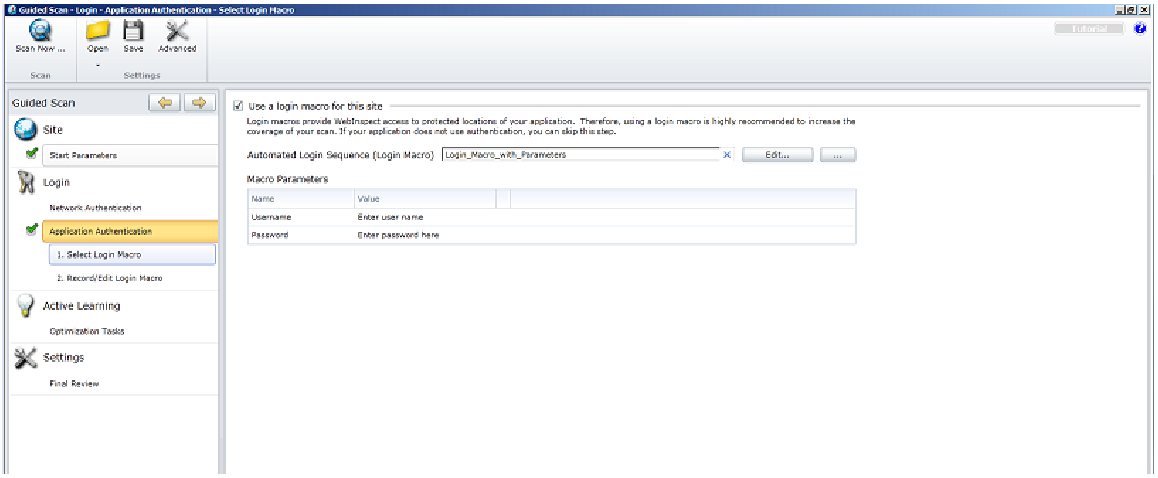
1. Select the macro step that contains the user name.
2. Place your mouse in the upper right corner of the step and click to open the Step Editor.
3. Click (expand) **Arguments**.
4. Highlight the entire contents of the **Value** text box, right-click the highlighted text, and select **Replace with a Parameter**.
5. On the Enter Parameter Name dialog box, select the parameter (**Username** in this example) from the **Select Parameter** list and click **OK**.

   The **Value** takes on the format of a parameter.
6. Select the macro step that contains the password.
7. Place your mouse in the upper right corner of the step and click to open the Step Editor.
8. Click (expand) **Arguments**.
9. Highlight the entire contents of the **Value** text box, right-click the highlighted text, and select **Replace with a Parameter**.
10. On the Enter Parameter Name dialog box, select the parameter (**Password** in this example) from the **Select Parameter** list and click **OK**.

    The **Value** takes on the format of a parameter.
11. Play the macro to verify that it logs in correctly.
12. If necessary, reopen the Parameters Editor and change the text in the **Value** text boxes to the default text that you want testers to see, as described in step 4 and step 7 under Task 1: Create Parameters.
13. Save the macro. (For a Guided Scan, click **Export**. For a Basic Scan in Fortify WebInspect or a Web Site Scan in Fortify WebInspect Enterprise, click **Save**.)

When you start a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise and select this macro, the parameters appear in the Macro Parameters table for a Guided Scan, or in the table below the name of the selected login macro in step 2 of a Basic Scan or a Web Site Scan. The tester simply replaces the parameters with a valid user name and password.

Image of Macro Parameters Table for a Guided Scan



☑ Use a login macro for this site

Login macros provide WebInspect access to protected locations of your application. Therefore, using a login macro is highly recommended to increase the coverage of your scan. If your application does not use authentication, you can skip this step.

Automated Login Sequence (Login Macro) Login_Macro_with_Parameters

Macro Parameters

Name	Value
Username	Enter user name
Password	Enter password here

When conducting a multi-user login scan, the multiple credentials configured for the scan in the Scan Settings: Authentication window are used. You are not prompted for input.

Using a URL Parameter

Task 1: Create Parameter

1. After recording and testing your macro, click **Parameters Editor** in the toolbar.

   The Parameters Editor opens.

2. Click + to add a parameter.
3. In the **Name** text box, enter a name for the parameter (for example: `StartURL`).
4. In the **Value** text box, enter the default text, Host Name, or URL (for example: `www.Placeholder.com`) that you want testers to see as the value for the parameter as they configure a scan. However, while you are developing the macro, for macro playback to succeed or for security reasons you might need to use a different, temporary default text, Host Name, or URL and change it to the default after you verify the macro near the end of this procedure.
5. Click **Close** to close the Parameters Editor.

Task 2: Assign Parameters to Steps

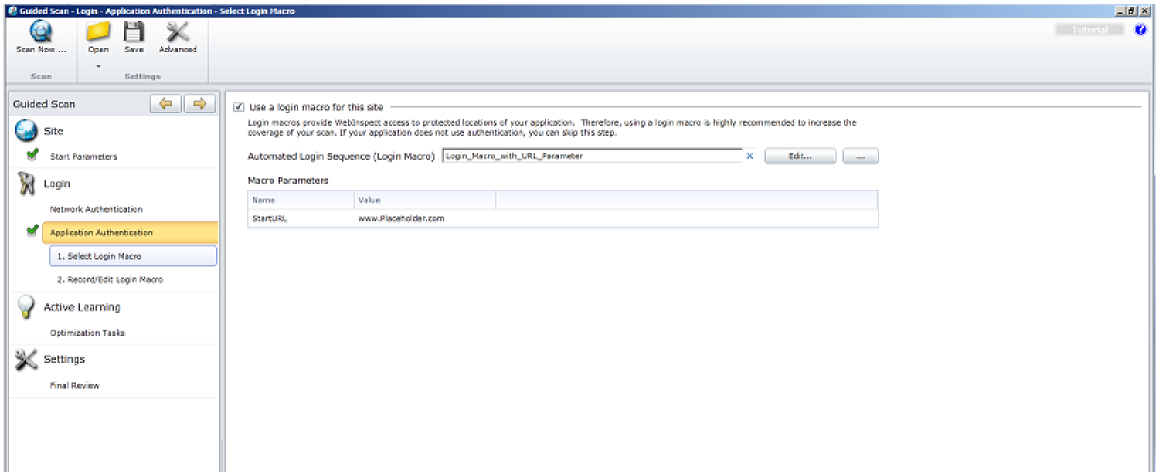
1. Select the macro step that contains the URL ("Navigate to...").
2. Place your mouse in the upper right corner of the step and click to open the Step Editor.
3. Click (expand) **Arguments**.
4. Highlight the entire contents of the **Location** text box, right-click the highlighted text, and select **Replace with a Parameter**.

5. On the Enter Parameter Name dialog box, select the parameter (**StartURL** in this example) from the **Select Parameter** list and click **OK**.

   The **Location** takes on the format of a parameter.
6. Play the macro to verify that it logs in correctly.
7. If necessary, reopen the Parameters Editor and change the text in the **Value** text box to the default text, Host Name, or URL that you want testers to see, as described in step 4 under Task 1: Create Parameter.
8. Save the macro. (For a Guided Scan, click **Export**. For a Basic Scan in Fortify WebInspect or a Web Site Scan in Fortify WebInspect Enterprise, click **Save**.)

When you start a Guided Scan, a Basic Scan in Fortify WebInspect, or a Web Site Scan in Fortify WebInspect Enterprise and select this macro, the parameters appear in the Macro Parameters table for a Guided Scan, or in the table below the name of the selected login macro in step 2 of a Basic Scan or a Web Site Scan. The tester either leaves the parameter unchanged (to access the original URL) or enters the URL of the target site.

Image of Macro Parameters Table for a Guided Scan



Guided Scan - Login - Application Authentication - Select Login Macro

☑ Use a login macro for this site

Login macros provide WebInspect access to protected locations of your application. Therefore, using a login macro is highly recommended to increase the coverage of your scan. If your application does not use authentication, you can skip this step.

Automated Login Sequence (Login Macro) Login_Macro_with_URL_Parameter

Macro Parameters

Name	Value
StartURL	www.Placeholder.com

How to Enhance Macros

There are a number of optional enhancements that can be added to macros.

Modify Steps

Modify step arguments and objects by selecting the desired step and expanding the options. This expands the step and allows you to modify the objects and properties. For a detailed list of the step structure, see "Toolbox" on page 212.

Insert Loops

A loop repeats a selected portion of the macro until certain criteria are met or for a specified number of times. To insert a loop, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **For loop** element to the desired location among the macro steps. For more information, see "How to Insert and Modify Loops" on page 212.

Insert If Blocks or If-else Blocks and Exit Steps

To conditionalize a portion of the macro, you can insert If or If-else blocks. To insert an If block, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **If block** element to the desired location among macro steps. To add an else condition, click the **Add else** link next to the If step title. For more details, see "Step Arguments" on page 216.

Exit steps cause a macro to exit the iteration or the entire macro. These can be used with If statements to exit a macro or iteration when a specified condition occurs. To insert an exit step, click **Toolbox**, click (expand) **Flow Control**, and click and drag the **Exit** element to the desired location among the macro steps.

Insert Comments

To insert comments into your macro, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Miscellaneous**, and click and drag the **Comment** element to the desired location among the macro steps.

Insert Catch Error Steps

"Catch error" steps are group steps that run their contents if the previous step contains an error. (To group steps, use **Ctrl** + click to select multiple steps, right-click any of them, and click **Group Steps**.) Additionally, the error is "caught" and is not returned. You can define catch error steps to catch any error, or a specific type of error. If there are two catch error steps in a row, they both apply to the same step. To insert a catch error step, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **Catch Error** element to the desired location among the macro steps.

Verify that an Object Exists

To verify that a string or object exists in the application, you can insert a verify step:

1. Click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Functions**, and click and drag the **Verify** element to the desired location among the macro steps.
2. Click the object in the verify step.
3. Select the object you want to verify.

Insert Generic Steps

You can insert a blank step and manually configure it. To insert a generic step, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Functions**, and click and drag the **Generic Object Action** element or the **Generic Browser Action** element to the desired location among the macro steps. Expand the step, and enter the desired step properties. Generic Object Actions perform an unspecified action on an object. Generic Browser Actions perform an unspecified action on the browser such as go back, reload, switch tabs, etc.

How to Debug Macros

This topic describes the basic steps involved in interactively debugging a macro.

View Replay Errors in Browser

If any steps failed during replay, they are marked with an error icon ⚠. Hover the mouse pointer over these icons to view descriptions of the errors.

Run the Macro Step by Step

The step-by-step replay allows you to view the sequence more slowly and in a controlled manner. To run the macro step by step, select the down arrow next to the **Replay** button in the right (macro steps) pane and select **Replay step by step**. Repeat this procedure after each step to continue the step-by-step replay.

Insert Breakpoints

Breakpoints instruct the macro to stop running during a replay when in interactive mode. They can be used to help debug your macro. To insert a breakpoint, select the desired step and click **Toggle breakpoints** in the macro steps toolbar (or right-click on the step and click **Toggle Breakpoint** in the popup menu).

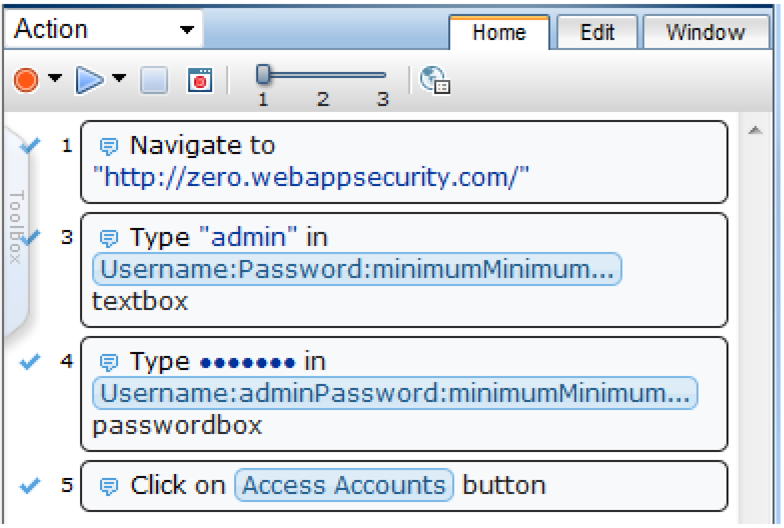
Modify Script Levels

As you record a macro, it assigns a level from 1 to 3 to each step. For example, a level 1 step is essential to the macro. A click step that occurs in an area of the application that has no effect is assigned to level 2. Mouse-over steps are generally considered unnecessary for the macro and are assigned to level 3.

Macro steps are displayed *and played* with the granularity specified as level 1, 2, or 3 in the **Script Level** slider in the macro steps toolbar at the top of the Home tab. The highest granularity is level 3—setting the slider to level 3 displays and plays back all the steps at levels 1, 2, and 3. Using higher granularity might be required for successful playback, but it can cause the macro to take longer to run. By default, the **Script Level** is set to 1.

To modify a macro's replay level, drag the **Script Level** slider in the macro steps toolbar to the desired level.

The following illustration shows a macro for which step 2 is hidden at **Script Level** 1.



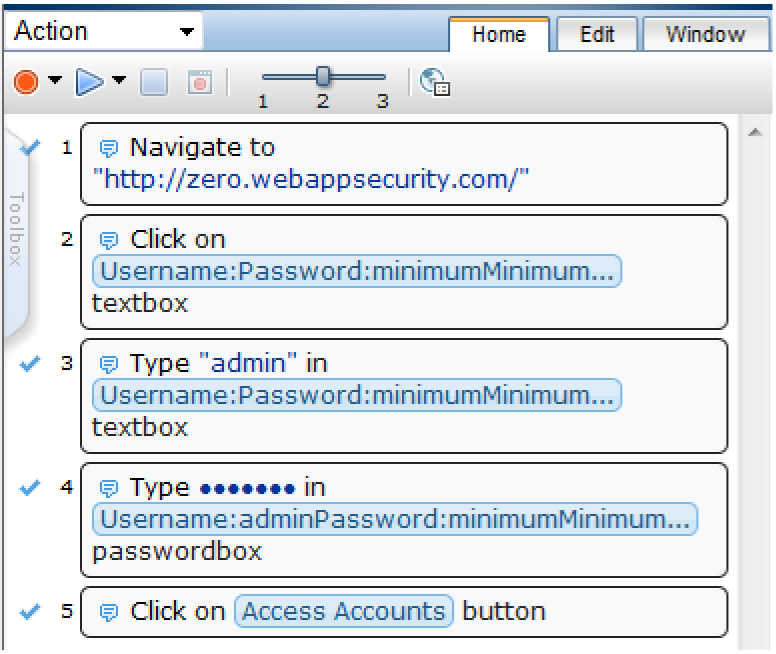
✓ 1 Navigate to "http://zero.webappsecurity.com/"

✓ 3 Type "admin" in Username:Password:minimumMinimum... textbox

✓ 4 Type ●●●●●●● in Username:adminPassword:minimumMinimum... passwordbox

✓ 5 Click on Access Accounts button

When the **Script Level** is changed to 2 as shown below (or if the **Script Level** were changed to 3), then macro step 2, which represents clicking in a text box and is assigned to **Script Level** 2, is also displayed and will run if the macro is replayed.



✓ 1 Navigate to "http://zero.webappsecurity.com/"

2 Click on Username:Password:minimumMinimum... textbox

✓ 3 Type "admin" in Username:Password:minimumMinimum... textbox

✓ 4 Type ●●●●●●● in Username:adminPassword:minimumMinimum... passwordbox

✓ 5 Click on Access Accounts button

In certain cases, you may want to manually change the level of a particular step, not the entire macro. For example, you may want to display and play a particular mouse-over step. To change the level of a step:

1. Place your mouse in the upper right corner of the step and click to open the Step Editor for the step.
2. Move the slider at the top of that step (to the right of the step number) to the desired level.

   If the step is part of a group step, both the group step and the individual step must be modified. (To group steps, use **Ctrl** + click to select multiple steps, right-click any of them, and click **Group Steps**.)

Insert Wait Steps

Wait steps cause the macro to pause for a specified amount of time before continuing with the next step. Wait for Object steps cause the macro to wait for a specified object to load before continuing with the next step. Wait steps begin after the End Event of the previous step is reached. This means that the previous step may continue to run after the wait step has been reached.

To insert a wait step, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Functions,** and click and drag the **Wait** element or the **Wait for Object** element to the desired location among the macro steps. Wait steps wait for a specified amount of time. Wait for Object steps wait until the specified object appears in the application. In Wait for Object steps, select the **Click to choose an object** button to select the target object in the application.

Disable/Enable Steps During Replay

To disable or re-enable a macro step during replay, place your mouse in the upper right corner of the step and click to open the Step Editor, and click the ☑ icon in the toolbar for the step. Alternatively, to disable or re-enable one or more steps, use **Ctrl** + click to select them, right-click one of the steps, and click **Disable Steps** or **Enable Steps** on the popup menu.

Disabled steps remain in the macro and can be re-enabled in the future, but are not played.

Make a Step Optional

Some steps can be made optional. An optional step is skipped during replay if its object is not found. To make a step optional, place your mouse in the upper right corner of the step and click to open the Step Editor, and click the icon in the toolbar for the step. To make a step non-optional again, click the icon again.

Play a Step

To play one step, place your mouse in the upper right corner of the step and click to open the Step Editor, and click the ▷ icon in the toolbar for the step.

Play From a Step to End of Macro

To start playback at one particular step and continue until the end of the macro, select the starting step, right-click on the step, and click **Play From This Step** on the popup menu.

How to Resolve Object Identification Issues

In dynamic websites, objects that have been recorded can often move or change content. Object identification presents one of the biggest challenges with recording and replaying Web 2.0 applications. This can cause the macro to lose the ability to locate the object.

The Web Macro Recorder includes sophisticated mechanisms to overcome this challenge, including the Highlight, Improve Object Identification, Replace, and Related Object options within steps that have objects. Using these options requires that you select an object in the application. For cases where various actions are required in the application to make the object visible, such as mouse over and mouse click, use the **Ctrl+Alt+F4** option to suspend the object-selection mode until you bring the object into view and press **Ctrl+Alt+F4** again to select the object.

When identifying objects for applications that were recorded in windows, use the Windows tab to make sure that the correct window is selected.

After you perform any of the changes, first replay the single failed step in question and then replay the entire macro again. This will help verify whether the change has solved the issue you encountered.

The following sections describe ways to resolve object identification issues.

Highlight an Object

Regardless of which method of object identification is used, place your mouse in the upper right corner of the step and click to open the Step Editor. Click (expand) **Object** and click the **Highlight** button to check at any time whether an object is visible in the application. If the object is found, it is temporarily surrounded by a flashing red outline. If the object is not found, an error message is displayed. The error could be an issue of pacing and timing, or that the correct page to find the object is not displayed.

Improve Object Identification

If the Highlight option fails, click the icon (with tooltip "Improve object identification") next to the selected ID Method for the object. This will let the Web Macro Recorder relearn the properties of the object and compare them to the properties learned during recording. Based on the differences, the necessary adjustments can be made. Depending on how dynamic the application is, you may need to use the Improve Object Identification function more than once.

Once you have done this, try replaying the step again to check whether the problem has been solved.

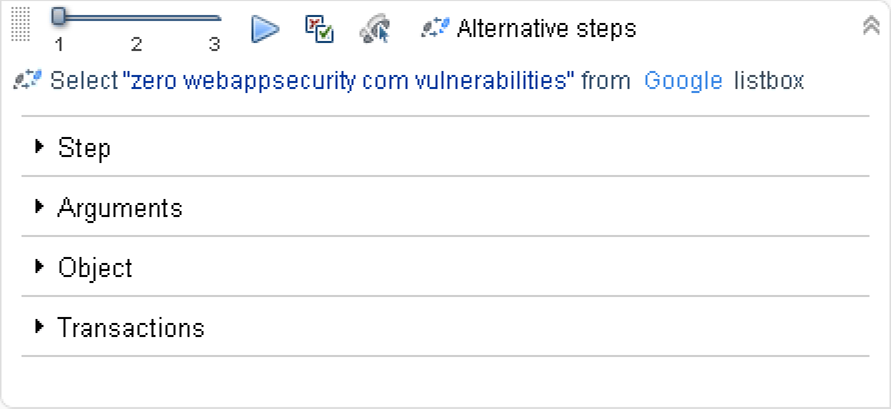
Consider Alternative Steps

Alternative steps allow you to view multiple ways to perform the same action in a step, where it is possible. You can modify the step for the best or most consistent macro performance, or for debugging purposes.

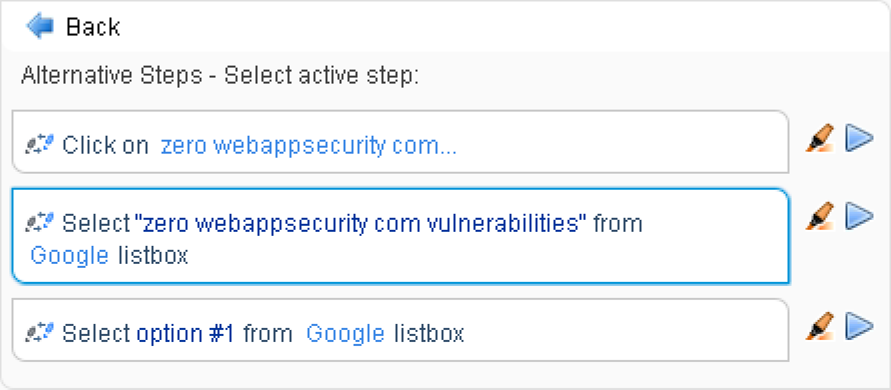
For example, you may be clicking on an option in a drop down list in which the text changes based on some value. If you try to click based on the text, the step may fail. If you use an alternative step that selects the item in the list based on the ordinal value of the option within the list, the click will succeed regardless of the text.

Steps that have alternative options are labeled with an alternative step icon on the left. Click the icon to view the alternative options for that step. (If the Step Editor is open, a button labelled **Alternative steps**, with the same icon, appears in the step's toolbar and performs the same function.)

The following screens show an example of alternative steps. After performing a Google search on "zero webappsecurity com vulnerabilities" and selecting "zero webappsecurity com vulnerabilities" as the first automatically generated item in the Google search box autocomplete, the step that has alternatives (expanded in the Step Editor) is shown below.


1
2
3
Alternative steps
Select "zero webappsecurity com vulnerabilities" from Google listbox
Step
Arguments
Object
Transactions

Clicking the icon displays the alternative steps shown below.


Back
Alternative Steps - Select active step:
Click on zero webappsecurity com...
Select "zero webappsecurity com vulnerabilities" from Google listbox
Select option #1 from Google listbox

The icon to the right of each alternative has the tooltip "Highlight the object in the AUT," where AUT means application under test. This performs the same highlighting function as described in the Highlight an object section in this topic, with the convenience of being able to highlight each alternative one at a time within the macro step.

Each alternative also has a Play icon so that you can confirm that using an alternative takes the appropriate action for the macro step.

Macro replay succeeded for each alternative.

Click the desired alternative to make it active, and click **Back** to return to normal display of the macro step using the alternative you selected. Replay the macro to test it.

Modify the Object Identification Method

You can modify the way the Web Macro Recorder identifies the object by modifying the object identification method (ID method) in the Object section of the Step Editor. The following options are available:

Option	Description
Automatic	The default and recommended object identification method. The Automatic method allows the Web Macro Recorder to use its internal advanced algorithms to locate the object. If this method does not successfully find the object during replay, click the 🔍 icon (with tooltip "Improve Object Identification") and replay the macro again.
XPath	If Automatic identification fails, even after using Improve Object Identification or Related Objects (described below), try using the XPath identification method. This method identifies the object based on an XPath expression that defines the object in the DOM tree. For example, if you need to select the first search result, regardless of the term being searched for, using XPath identification may help. Click the drop-down arrow next to the **XPath** edit box to select a suggested XPath for the object. You can click the popup **Edit** button at the right end of the **XPath** edit box to open the XPath Editor and edit the suggested XPath. For the XPath ID method, the tooltip for the 🔍 icon changes to "Regenerate expression." When you click the icon, you can select an object in the interface and thereby create its associated XPath.
JavaScript	JavaScript code that returns an object. For example: document.getElementById ("SearchButton") returns an element that has a DOM ID attribute of "SearchButton." Using the JavaScript identification method, you can write JavaScript code that references the returned document and you can use CSS selectors and other standard functions. For example, the page returned by the server contains multiple links with the same "title" attribute (search results) and we want the script to randomly click on one of the available links. Object identification for this case, using the JavaScript identification method, may look similar to the following:

Option	Description

	`var my_results = document.querySelectorAll('a [title="SearchResult"]');`
	`random(my_results);`

Modify the Macro Timing

Sometimes objects may not be found because of timing and synchronization issues. For example, the macro may be looking for an object that was in the application, but the macro replayed too quickly and already progressed to another page. If you suspect that the object is not being found because of a timing or synchronization issue, you can insert Wait steps. For more information, see "How to Debug Macros" on page 205.

Relate Objects to Other Objects

If the preceding options do not solve the issue, try using the Related Objects option.

If an object becomes difficult to identify on its own, you can label the object based on a different, more stable object. For example, you can select an object that is not dynamic and "relate it" to the target object. Relations are defined visually, relating objects according to their distance in pixels from other objects. Relations are defined per ID method, per object. If more than one relation is defined for an ID method of a given object, both relations must locate the same object for the step to pass. To use this function, place your mouse in the upper right corner of the step and click to open the Step Editor, click (expand) **Object**, click (expand) **Related Objects**, and click +. Follow the directions to create a relation. Verify that it has worked by highlighting both the object and its related object.

Tips:

- Use this feature only if other identification methods have failed, as it may be more resource intensive.
- Use the minimum search area to improve performance.
- Related Objects are sensitive to window sizing. Resizing may alter object positions and relationships. This should be taken into account.
- Each identification method (Automatic, XPath, and JavaScript) has its own set of related objects. These related objects are not shared among identification methods.
- If several relations exist, they all need to be found in order for the identification to succeed.

Replace an Object

If you selected the wrong object during recording, or an object has permanently changed, you can replace it with a different object without replacing the step. This effectively resets the step, deleting changes made to the original step such as relations. Place your mouse in the upper right corner of the step and click to open the Step Editor, click (expand) **Object**, and click **Replace** . Select the new object and replay the macro.

Using this option tells the macro recorder that the object currently referenced in the step is incorrect. The macro recorder will remove any current knowledge of the object and learn the object you select. Therefore, you should only use the **Replace** option if the object you used during recording was the wrong one.

How to Insert and Modify Loops

Loops repeat selected portions of the macro until certain criteria are met or for a specified number of iterations. You can insert loops and loop modifiers from the **Flow Control** section of the **Toolbox**.

"For" Loops

"For" loops perform the steps surrounded by the loop until the end condition is met or the code reaches a break statement. Loop arguments use JavaScript syntax. To insert a For loop, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **For loop** element to the desired location among the macro steps.

"Break" Statements

Break statements indicate that the current loop should end immediately. For example, if a Break statement is encountered in the second of five iterations in a For loop, the loop will end immediately without completing the remaining iterations. To insert a Break statement, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **Break** element to the desired location among the macro steps.

"Continue" Statements

Continue statements indicate that the current loop iteration should end immediately. The loop condition is then checked to see if the entire loop should end as well. For example, if a Continue statement is encountered in the second of five iterations in a For loop, the second iteration will end immediately and the third iteration will begin. To insert a Continue statement, click the **Toolbox** vertical tab on the left side of the macro steps pane, click (expand) **Flow Control**, and click and drag the **Continue** element to the desired location among the macro steps.

Toolbox

The toolbox, a vertical tab on the left side of the macro steps pane, enables you to add steps to macros. When you click **Toolbox** and click (expand) one of the headings such as **Functions**, you can click and drag a particular element such as **Verify** to add it to the macro steps.

You can click and drag **Toolbox** to move the toolbox up or down. To close the toolbox, click **Toolbox** again.

User interface elements are described in the following table.

UI Element	Description
Functions	**Verify**. Verify that an object exists in the application. **Wait.** Wait for a specified number of seconds before continuing with the next step. **Wait for Object.** Wait for an object to load before continuing with the next step. **Generic Object Action** or **Generic Browser Action**. Blank steps that can be inserted and manually configured.
Flow Control	**For Loop**. A logical structure that repeats the steps contained in the loop a specified number of times. **If Block.** A logical structure that runs the steps contained in the block if the condition is met. • **Add else.** Click the **Add else** link to add an **else** section to your **If** block. If the condition is not met, the steps included in the **else** section run. • **Remove else.** Removes the **else** section from the **If** block. **Note:** If the **else** section contains steps and you click **Remove else**, the steps are deleted. Copy and paste them into the main body of your macro to save them. **Break.** Causes the loop to end immediately without completing the current or remaining iterations. **Continue**. Causes the current loop iteration to end immediately. The macro continues with the next iteration. **Catch Error.** Catches an error in the step immediately preceding and runs the contents of the catch error step. For more information, see "How to Enhance Macros" on page 203. **Exit.** Exits the iteration or the entire macro depending on the specified setting.
Miscellaneous	**Evaluate JavaScript**. Runs the JavaScript code contained in the step. **Evaluate JS on Object.** Runs the JavaScript code contained in the step after the specified object is loaded in the application. **Comment.** A blank step that allows you to write comments in your macro.

UI Element	Description
Composite Steps	**Answer Security Questions.** Allows you to select the interface object (usually a label) that asks a security question and the interface object (usually a text box) where the user provides the answer. Then you specify the text of the question and the answer.

General Settings

Click **General Settings** in the toolbar of the macro steps pane to open the *General Settings* dialog.

Snapshot Generation

A snapshot is an image of the browser taken at the times specified by the following options:

- **Recording snapshots generation.** Select **Never** (the default) or **Always**. During macro recording, any snapshots that are taken are saved in the same folder as the macro.
- **Replay snapshots generation.** Select **Never**, **On error** (the default), or **Always**. During a scan, any snapshots that are taken are saved in the log directory.

Replay Options

Specify the following options:

- **Maximum time for object-not-found (seconds).** Specify the maximum time (in seconds) that the macro recorder will wait for the target object of a replay step to appear.
- **Inter-step interval (milliseconds).** Specify the minimum interval (in milliseconds) between steps.
- **End-of-network identificationtimeout (milliseconds).** Specify the timeout (in milliseconds). The end-of-network for a step is recognized when the specified time has elapsed with no network activity.
- **Clear image cache per user.** If you select this option, the image cache will be cleared during replay.

Log Level

Select one of the following options:

- **Standard logging.** Log only warnings and high-level informational messages.
- **Extended logging.** Log low-level messages, warnings, and high-level informational messages.

Logout Condition

In the **XPath depth** option, specify the depth used for XPath in logout detection by element. The depth determines the number of xpath locators (parents) from the element up to its ancestors.

An element can be located (found) in a page using a path to its location. For example, in the following HTML, to locate <div class="painter" id="painterId">, the search can use the following: find body, then find div with id painterId, or the search can use find body-> then find second div.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html lang="en">
    <head>
        <title>colors</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />

    </head>
        <body>
            <div class="container">

                <div class="box">
                    <div class="caret" id="red">
                        <span></span>
                    </div>
                </div>
                <div class="number" id="redNumber">0</div>
                <div class="box">
                    <div class="caret" id="green">
                        <span></span>
                    </div>
                </div>
                <div class="number" id="greenNumber">0</div>
                <div class="box">
                    <div class="caret" id="blue">
                        <span></span>
                    </div>
                </div>
                <div class="number" id="blueNumber">0</div>
            </div>
            <div class="painter" id="painterId">Color</div>
            <script type="text/javascript" language="javascript">initialize
();</script>
        </body>
    </html>
```

So, when searching through larger html files with more complex structures, the process can use either a rigid full xpath, or a loose short xpath. The default value is 3.

Encryption

If you select the **Encrypt Macro** option, the entire macro file is encrypted when saved. Otherwise, the file is saved in plain text, which exposes user names and passwords. This option is selected (ON) by default.

Step Arguments

The following table displays the step arguments categorized by role. Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code.

Role	Action	Arguments
element	Evaluate JavaScript	**Code:** JavaScript code
element	Mouse Actions: Mouse Down, Mouse Up, Mouse Over, Click, Double Click	• **Button:** The mouse button that is clicked. • **X Coordinate:** The offset location of the action relative to the upper left corner of the object. This number must be positive. If not specified, the default is the center of the object. • **Y Coordinate:** The offset location of the action relative to the upper left corner of the object. If not specified, the default is the center of the object. • **Ctrl Key:** Whether or not this key is pressed during the action. • **Alt Key:** Whether or not this key is pressed during the action. • **Shift Key:** Whether or not this key is pressed during the action.
element	Drag	• **Button:** The mouse button that is clicked. • **X Offset:** The number of pixels to drag the object on the x axis. A positive number indicates a drag to the right. • **Y Offset:** The number of pixels to drag the object on the y axis. A positive number indicates a drag down.

Role	Action	Arguments
		• **Path:** List of coordinates representing user drag path. Do not modify this argument.
element	Drag To	• **Target Object:** The step object is dragged to this target object. • **X Offset:** The offset from the top left of the target object in the x axis. This number must be positive. • **Y Offset:** The offset from the top left of the target object in the y axis. This number must be positive.
element	Get Property	• **Property.** The property whose value will be stored in the specified variable. The list of properties available depends on all the roles of the object. • The default properties available for all objects are: • **Visible text.** The visible text of the item, corresponding to the DOM textContent property. • **All text.** The entire text of the item, corresponding to the DOM textContent property. • **Inner HTML.** The inner html markup of the object, corresponding to the DOM innerHTML property. • **Variable.** The name of the variable in which to store the specified property value.
element	Verify	• **Value:** The string or number to verify. • **Property:** The object property whose value will be verified. The list of properties available to verify depends on all the roles of the object. • The default properties available for verification on all objects are: • **Visible text.** Items that are visible in the application. • **All text.** Items that are in the application but are not necessarily visible. Items in this category are contained in DOM property textContent. • **Inner HTML.** Items contained in the DOM property innerHTML.

Role	Action	Arguments
		• **Condition:** The relationship between the value and property arguments.
element	Wait for Property	• **Value:** The value of the specified property that the step will wait for, before the step passes. • **Property:** The object property whose value the macro will wait for. The list of properties available for which to wait depends on all the roles of the object. • The default properties available for all objects are: • **Visible text.** Items that are visible in the application. • **All text.** Items that are in the application but are not necessarily visible. Items in this category are contained in DOM property textContent. • **Inner HTML.** Items contained in the DOM property innerHTML. • **Condition:** The relationship between the value and property arguments.
focusable	Press Key	**Key name:** Enter or Space.
text box	Type	• **Value:** What is typed. • **Clear:** Clear the text box before typing. The default is true. • **Typing Interval:** The average time in milliseconds between keystrokes.
checkbox	Set	**Checked:** Set the checkbox to either checked (T) or unchecked (F).
listbox	Select	• **Text:** The selected string. • **Ordinal:** The order of the selected item in the list. If the text argument is also specified, then this argument refers to the instance of the specified text value in the listbox. An ordinal of 0 generates a random value.
radiogroup	Select	• **Text:** The selected string. • **Ordinal:** The order of the selected item in the list. If the text argument is also specified, then this argument refers to the instance of the specified text value in the listbox. An ordinal

Role	Action	Arguments
		of 0 generates a random value.
filebox	Set	**Path:** The selected path.
slider	Set	**Value:** The value that the slider is set to.
datepicker	Set Day	**Day:** An integer between 1 and 31 representing the day of the month.
browser	Activate	**Ordinal:** Defined as an integer. Moves the specified browser window to the foreground.
browser	Activate Tab	**Ordinal:** Which tab (integer) to activate.
browser	Close Tab	**Ordinal:** Which tab (integer) to close.
browser	Add Tab	**Location:** The URL to navigate to in the newly opened tab.
browser	Navigate	**Location:** The URL to navigate to.
browser	Go Back	**Count:** The number of pages to go back.
browser	Go Forward	**Count:** The number of pages to go forward.
browser	Resize	• **Width:** The new width. Leaving this blank means do not resize the width. • **Height:** The new height. Leaving this blank means do not resize the height.
browser	Scroll	• **X Coordinate:** The new x coordinate. Leaving this blank means do not scroll along the x axis. • **Y Coordinate:** The new y coordinate. Leaving this blank means do not scroll along the y axis.
browser	Dialog - Confirm	**Button:** Ok or Cancel.
browser	Dialog Prompt	• **Value:** The string to enter. • **Button:** Ok or Cancel.
browser	Dialog - Authentication	• **Username:** The username to enter. • **Password:** The password to enter. • **Domain:** The domain to enter.

Role	Action	Arguments
		• **Button:** Ok or Cancel.
browser	Verify	• **Value:** The value of the property to verify. • **Property:** The property to verify. You can verify the following properties of a browser object: • **Title.** The title of the browser window. • **Location.** The location of the browser window. • **Condition.** The relationship between the value and property arguments.

See Also

"How to Enhance Macros" on page 203

Chapter 18: Web Macro Recorder with Macro Engine 5.0

Fortify WebInspect includes the Web Macro Recorder tool, which was designed with TruClient technology. It uses event-based functionality and Firefox browser technology to record macros.

Important! This version of Web Macro Recorder is provided as a technology preview.

Technology Preview

Technology preview features are currently unsupported, may not be functionally complete, and are not suitable for deployment in production. However, these features are provided as a courtesy and the primary objective is for the feature to gain wider exposure with the goal of full support in the future.

About this Technology Preview

This technology preview version of the Web Macro Recorder is available if **Macro Engine 5.0** is selected as the Default Web Macro Recorder in Application Settings: General in Fortify WebInspect.

Additionally, this version of the Web Macro Recorder:

- Is accessible from the Basic Scan Wizard in Fortify WebInspect only
- Is not accessible from nor compatible with the Guided Scan Wizard
- Is not accessible from Fortify WebInspect Enterprise

About Macros

A login macro is a recording of the events that occur when you access and log in to a website. You can subsequently instruct Fortify WebInspect to begin a scan using this recording. A workflow macro is a recording of specific URLs on a site. For more information, see "Login Macros" on the next page and "Workflow Macros" on page 223.

Firefox Technology

The Web Macro Recorder uses underlying Firefox browser technology to record and play macros.

Web Macro Recorder Limitations

The Web Macro Recorder does not support the recording of Flash or Silverlight applications.

The TruClient technology used in the Web Macro Recorder is an adaptation of the Ajax TruClient technology originally developed for use with Micro Focus LoadRunner and Micro Focus Performance Center. It does not incorporate or support all the capabilities of the fully-featured version in those products.

Cookie Headers in Macros

When you play a macro, Fortify WebInspect does not send any cookie headers that may have been incorporated in the recorded macro.

URLs in Macros

If a URL is in a macro, the request is always sent when the macro is played, regardless of any exclusion rules in scan settings.

Login Macros

A login macro is a recording of the activity that is required to access and log in to a website or web application, typically by entering a user name and password and clicking a button such as Log In or Log On. When you configure a scan, you usually specify a previously recorded login macro or record a new one at the time for the scan to use.

Logout Conditions

To prevent the scan from terminating prematurely if Fortify WebInspect gets logged out of your application, a login macro should also specify at least one logout condition that definitively indicates that a logout has occurred. During a scan, Fortify WebInspect can get logged out for a variety of reasons, including:

- Normal logout driven by the target site
- An error condition in the target site such as a timeout
- An error in the macro itself, such as an invalid parameter

Specifying a logout condition as part of the login macro makes it unnecessary for users to manually log back in, perhaps repeatedly, when unexpected logouts occur during a scan. When scanning a site, the scanner analyzes every target site response to determine the state. If the scanner determines at any time that it is logged out, it runs the login macro to log back in, and then it resumes crawling or auditing the site at the point where the logout occurred.

You can specify multiple logout conditions, and if any of them are met, Fortify WebInspect plays the login macro to log the scanner back in and resume the scan where it left off.

See Also

"Working with Logout Conditions" on page 238

Workflow Macros

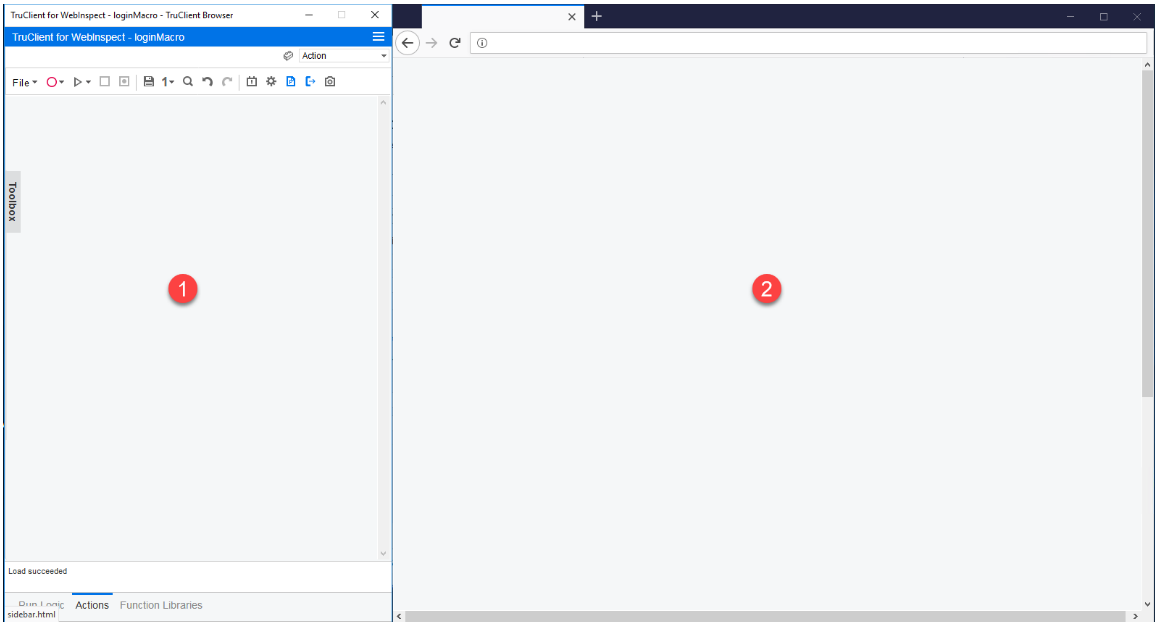
A workflow macro is a recording of the specific URLs to which you manually navigate on a site. When you configure a Basic Scan in Fortify WebInspect, you specify a previously recorded workflow macro or record a new one at the time for the scan to use. Fortify WebInspect audits only the URLs that are recorded in the workflow macro and does not follow any hyperlinks encountered during the audit. This type of macro is used most often to focus on a particular subsection of an application. In terms of the macro recording process, the essential differences from login macros are that:

- Workflow macros include only the specific URLs to which a user navigated while recording them. Workflow macros access only those URLs upon replay.
- Workflow macros do not require logout conditions.

> **Note:** If your website requires authentication, do not record login steps in a workflow macro. Instead, record a separate login macro to log in to your website. For more information, see "Login Macros" on the previous page.

Understanding the User Interface

The Web Macro Recorder opens with two windows side by side as shown in the following image.


TruClient for WebInspect - loginMacro - TruClient Browser
TruClient for WebInspect - loginMacro
Action
File
1
Toolbox
1
2
Run Logic
Actions
Function Libraries
sidebar.html

Load succeeded

The following table describes the two windows.

Window	Description
1	The TruClient sidebar window. Use this window to control the recording and editing functions.
2	The Firefox browser window. Use this window to access your website.

TruClient Sidebar Menu Toolbar

The following table describes the menu toolbar, which is available at the top of the TruClient sidebar.

Icon	Name	Description
File ▾	**File**	Use to open existing, create new, and save copies of macros and script files.
○ ▾	**Record**	Starts recording the macro. Additionally, you can use the arrow to specify whether to record before, into, or after the selected step.
▷ ▾	**Replay**	Replays (or resumes replay of) the macro. Additionally, you can use the arrow to specify whether to play the selected step only, or to run the script step by step. Running the script step by step pauses the replay after each step.
□	**Stop**	Stops recording or replaying the macro.
⊡	**Toggle Breakpoint**	Toggles a breakpoint on the selected step. For more information, see "Using Breakpoints" on page 274.
🖫	**Save**	Saves the macro.
1 ▾	**Step Level**	Modifies the script levels that are visible and replayed in the script. For more information, see "Modifying the Macro Replay Level" on page 236.
🔍	**Find**	Lists the steps that match the text string specified, or goes to a specific step number.
↶ ↷	**Undo/Redo**	Reverses your last action or restores your original change.
📅	**Event Handler Editor**	Opens the Event Handler Editor dialog box. For more information, see "Using the Event Handler Editor" on page 237.

Icon	Name	Description
⚙	**General Settings**	Opens the General Settings dialog box. For more information, see "Configuring Settings" on page 283.
P	**Edit Parameters**	Set parameter values. For more information, see "Parameters Dialog" on page 243.
⇥	**Edit logout conditions**	Opens the Logout Condition Editor. For more information, see "Working with Logout Conditions" on page 238.
📷	**Snapshot view**	Opens the Snapshot Viewer. For more information, see "Viewing Snapshots" on page 242.

Manage Actions

The **Manage Actions** icon (📦) is located above the TruClient sidebar menu toolbar. It opens the Actions dialog box. For more information, see "Working with Actions" on page 240.

Context Menu

Select a step in the TruClient sidebar and right click to display the context menu. The following table describes the context menu options.

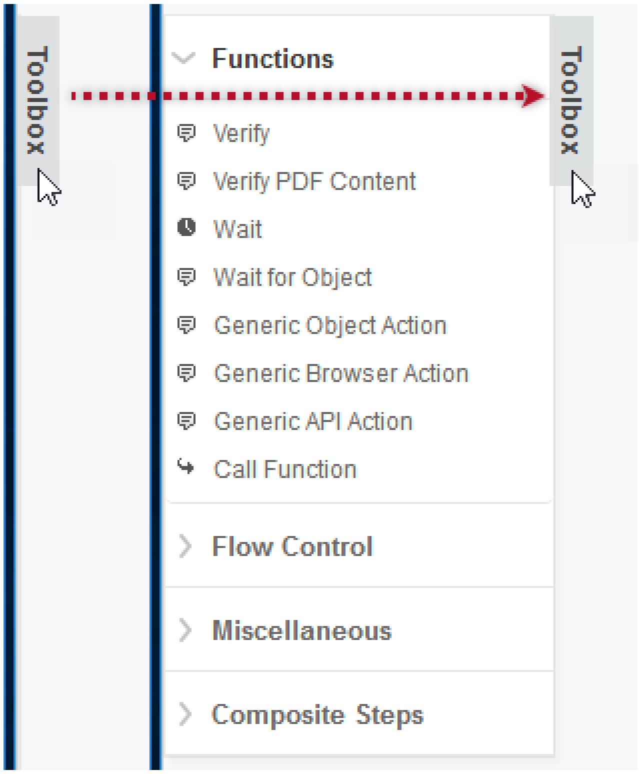
Menu Option	Description
Play This Step	Replays the selected step only.
Play From This Step	Replays from the selected step. You cannot use **Play From This Step** if the target step: • Is located in an action that is not part of the run logic. • Is inside a For loop or If block. • Is a Catch Error step. • Acts on an Web object that is not available on the current Web page.
Play Until This Step	Replays from the beginning and stops before the selected step.
Record > Before step	Inserts the next set of recorded steps before the selected step.
Record > Into	Inserts the next set of recorded steps into the selected step.

Menu Option	Description
step	
Record > After step	Inserts the next set of recorded steps after the selected step.
Toggle Breakpoint	Inserts or removes a breakpoint on the selected step.
Group Steps	Groups multiple steps together as a single step.
Group Into	Groups multiple steps into: • **Action** - A group of steps that you define as a new or existing action. • **If Clause** - A logical structure that controls the flow of your script. • **For Loop Clause** - A logical structure that repeats the steps contained in the loop a specified number of times. • **New Function** - A group of steps, such as a login, that you define as a function.
Ungroup Steps	Reverts grouped step into multiple steps.
Cut	Cuts the selected step from the macro.
Copy	Copies the selected step in the macro.
Paste	Pastes the copied step into the macro.
Export	Copies the selected steps in a macro to paste into another macro.
Import	Pastes the steps that have been exported into a second script.
Delete	Deletes a step from the macro.
Disable/Enable	Toggles between disabling or enabling a step during replay.
Edit Step	Expands the step to display the step, argument, and transaction properties.
Fold All Steps	Minimizes all steps and groups.
Unfold All Steps	Displays all steps and groups.
Reset Auto End Event	Enables you to reset the selected step or steps to **Automatic: Not Yet Set**.

Menu Option	Description
Change Object Identification Method	Enables you to change the object identification method to: • Automatic • XPath • JavaScript • Descriptors

Understanding the Toolbox

The toolbox is a vertical tab on the left side of the TruClient sidebar window. When you click the tab, it expands to display a list of options that enable you to add steps to macros.



Using the Toolbox

When you click **Toolbox** and click one of the headings such as **Functions**, you can click and drag a particular element such as **Verify** to add it to the macro steps.

You can click and drag **Toolbox** to move the toolbox up or down. To close the toolbox, click **Toolbox** again.

Function Options

Function options are described in the following table.

Option	Description
Verify	Verify that an object exists in the application.
Verify PDF Content	Verify the content of a PDF document during macro execution.
Wait	Wait for a specified number of seconds before continuing with the next step.
Wait for Object	Wait for an object to load before continuing with the next step.
Generic Object Action, **Generic Browser Action**, or **Generic API Action**	Blank steps that can be inserted and manually configured. For API argument details, see the specific API in the API Help at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/reference/functions.htm.
Call Function	Insert a custom function in the macro. For more information, see TruClient functions and function libraries at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/_tc_c_step_functions.htm.

Flow Control Options

Flow control options are described in the following table.

Option	Description
For Loop	A logical structure that repeats the steps contained in the loop a specified number of times. For more information, see "Inserting Loops and Loop Modifiers" on page 269.
If Block	A logical structure that runs the steps contained in the block if the condition is met. • **Add else** – Click the **Add else** link to add an **Else** section to your **If** block. If the condition is not met, the steps included in the **Else** section run. • **Remove else** – Removes the **Else** section from the **If** block. **Note:** The else sections apply to all If types (If Block, If Exists, If Verify,

Option	Description
	and If Browser). If the **Else** section contains steps and you click **Remove else**, the steps are deleted. Copy and paste them into the main body of your macro to save them. For more information, see "Inserting If Blocks, If-else Blocks, and Exit Steps" on page 270.
If Verify	A combination of "If Block" and "Verify," a logical structure that runs the steps contained in the block if the condition on a property of the selected object is met.
If Exists	A logical structure that runs the steps contained in the block if the selected object exists in the application.
Break	Causes the loop to end immediately without completing the current or remaining iterations.
Continue	Causes the current loop iteration to end immediately. The macro continues with the next iteration.
Catch Error	Catches an error in the step immediately preceding and runs the contents of the catch error step. For more information, see "Inserting Catch Error Steps" on page 271.
Exit	Exits the iteration or the entire macro, depending on the specified setting.

Miscellaneous Options

Miscellaneous options are described in the following table.

Option	Description
Evaluate JavaScript	Runs the JavaScript code contained in the step.
Evaluate JS on Object	Runs the JavaScript code contained in the step after the specified object is loaded in the application.
Comment	A blank step that allows you to write comments in your macro.

Composite Steps Option

The **Answer Security Questions** option allows you to select the interface object (usually a label) that asks a security question and the interface object (usually a text box) where the user provides the answer. Then you specify the text of the question and the answer.

Working with Macros in Basic Scan

You can access the TruClient Web Macro Recorder when configuring a Basic Scan in Fortify WebInspect.

Creating or Using a Login Macro in a Basic Scan

To create or use a login macro when you are completing Step 2 of a Basic Scan:

1. Select the **Site Authentication** option to use a login macro, and then do one of the following:
    - To record a new login macro, click **Record**.
    - To select (and optionally edit) an existing login macro to use in the scan, click the browse button to navigate to the macro.

      > **Note:** If the selected login macro uses parameters, a table displays those parameters when you select that login macro. Edit the parameter values as needed.

    The TruClient Web Macro Recorder opens with two windows in view: the TruClient browser sidebar window and the Firefox browser window. For more information, see "Understanding the User Interface" on page 223.
2. Proceed as follows:
    - If you are recording a login macro, follow the steps described in "Recording a Macro" below.
    - If you are editing a login macro, follow the steps described in "Editing a Macro" on page 232.
    - If you are selecting an existing login macro without editing it, after selecting a macro, complete Step 2 of the Basic Scan, and then continue configuring the scan.

Recording a Macro

When recording a macro, use the TruClient sidebar to control the recording functions and the Firefox browser to access your website.

Recording a Login Macro

This procedure describes how to record a basic login macro. For information about challenge-response login macros, see "Challenge-Response Authentication" on page 232 and "Recording a Macro for

Challenge-Response Logins" on page 233.

To record a login macro:

1. In the Firefox browser, navigate to the start URL for your website.
2. In the TruClient sidebar, click the **Record** icon (○▾).
3. In the Firefox browser, navigate to the login form and log in to the application.
4. After you log in, click the **Stop** icon (□) in the TruClient sidebar, but do not log out.
5. In the TruClient sidebar, click the **Play** icon (▷ ▾) to verify that your macro logs in correctly.
6. Did the macro log in correctly?
   - If *yes*, the TruClient sidebar prompts you to select an object to indicate a successful login. Proceed to the next step.
   - If *no*, click **File > New**. If prompted, do not save the macro. Return to Step 1.
7. In the Firefox browser, identify an object that appears only after successful login.

   A wait action for the selected object is added to the recorded steps.

   The Web Macro Recorder attempts to automatically detect a logout condition. For information about how to add or edit logout conditions later, see "Working with Logout Conditions" on page 238.
8. Click the **Save** icon (💾) to save the macro.

To add options to the login macro, see "Enhancing Macros" on page 268.

Recording a Workflow Macro

To record a workflow macro:

1. In the Firefox browser, navigate to the start URL for your workflow.
2. In the TruClient sidebar, click the **Record** icon (○▾).
3. In the Firefox browser, navigate to the pages you want to record in the macro.
4. After you record your navigation, click the **Stop** icon (□) in the TruClient sidebar.
5. Do one of the following:
   - To verify that your navigation was recorded correctly, click the **Play** icon (▷ ▾) in the TruClient sidebar.
   - To add options from the Toolbox to your recorded navigation, click **Toolbox**. For more information, see "Understanding the Toolbox" on page 227.
6. When you have finished, click the **Save** icon (💾) to save the macro.

Editing a Macro

As you edit a macro, you use the TruClient sidebar to add or edit the recorded steps and the Firefox browser to access your website. For more information, see "Understanding the User Interface" on page 223.

To edit a macro:

1. In the TruClient sidebar, click **File > Open**.
2. Add or edit steps in the macro. For more information, see "Enhancing Macros" on page 268 and "Debugging Macros" on page 273.
3. Click the save icon (💾) to save the macro.

Challenge-Response Authentication

Challenge-response authentication is a family of protocols in which the server presents a question (the challenge) and the client must provide a valid answer (the response). In the simplest example, the challenge asks for a password and the valid response is the correct password.

Multiple Challenges

Some websites present multiple challenges to the user. Typically, when a user first registers with a website, the site presents a list of questions to which the user provides answers that will be used for subsequent authentication. For example:

- What is your favorite color?
- What was the name of your first pet?
- In what town or city were your born?
- What was the make of your first automobile?

When the user later attempts to log in, the website presents two or more of these challenges.

Groups of Challenges

Some sites also create groups of challenges, and present questions from the groups on each new login attempt, as demonstrated in the following example.

When registering for the example website, the user is asked to provide answers to nine questions, which are arranged into three groups of three questions each, as follows.

Group 1

Q: What is your quest? A: happiness
Q: What is your name? A: Smith
Q: What is your favorite color A: blue

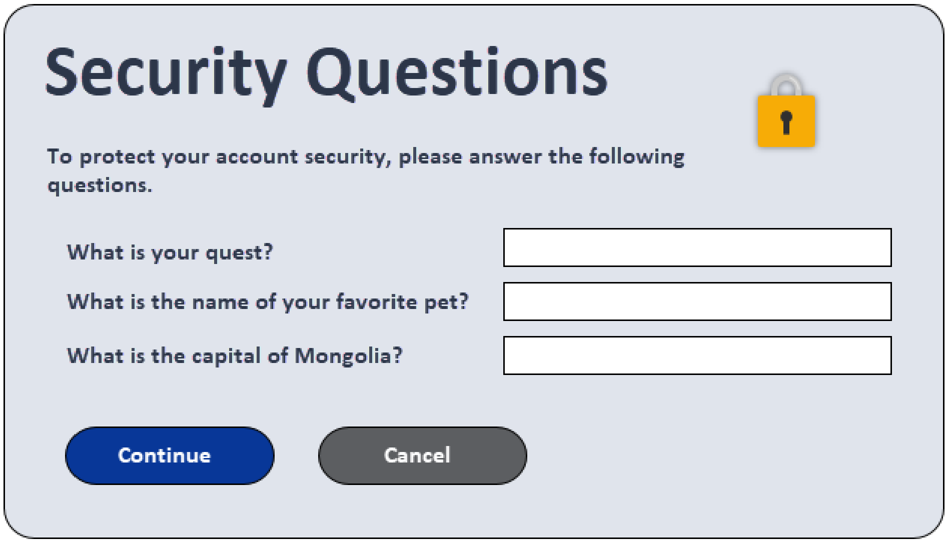
Group 2

Q: What is the name of your favorite pet? A: Rusty
Q: What is your mother's maiden name? A: Jones
Q: In what state were you born? A: Delaware

Group 3

Q: What is the capital of Mongolia? A: Ulaanbaatar
Q: What is the name of a sea bird? A: Albatross
Q: What is your paternal grandmother's first name? A: Esther

The login page might look like this (using the first question from each group):



Security Questions

To protect your account security, please answer the following questions.

What is your quest?

What is the name of your favorite pet?

What is the capital of Mongolia?

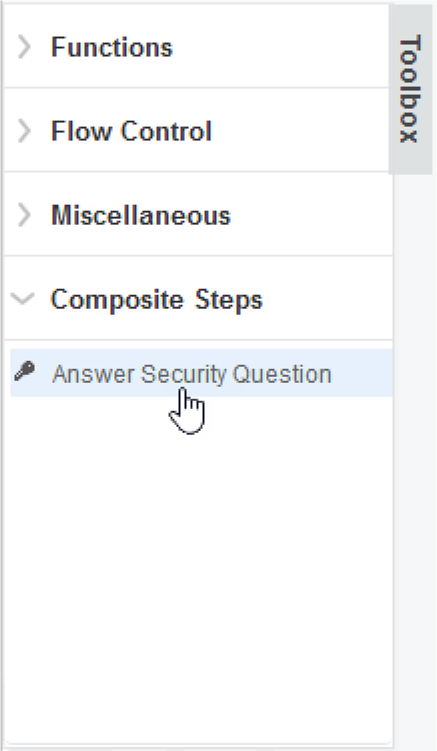
Recording a Macro for Challenge-Response Logins

When recording a macro for a challenge-response type of login, you must know all possible question-and-answer combinations, even if only a subset of those combinations might be presented during any one login. You enter these combinations manually, as special steps as you record a macro.

At the point where the target site asks the challenge questions, usually after logging in with username and password credentials, use the following procedure to manually create the required steps for the set of questions:

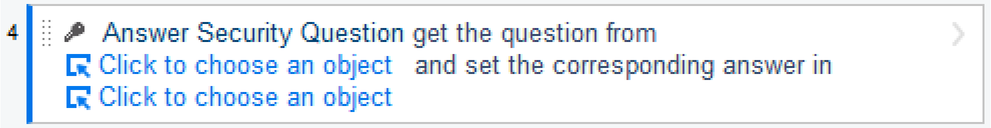
1. While recording the macro, click the **Stop** icon (□) in the TruClient sidebar.
2. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.

3. Click (expand) **Composite Steps**.



4. Click and drag the **Answer Security Questions** element into the recorded steps.

   A new step is created.



4 Answer Security Question get the question from Click to choose an object and set the corresponding answer in Click to choose an object

5. Click the first **Click to choose an object** link in the new step and then, in the Firefox browser window, click the object representing the question (usually a label).
6. Click the second **Click to choose an object** link in the new step and then, in the Firefox browser window, click the object representing the answer (usually a text box).
7. In the TruClient sidebar, hover your mouse over the Answer Security Questions step.

   The Step Editor icon becomes visible.

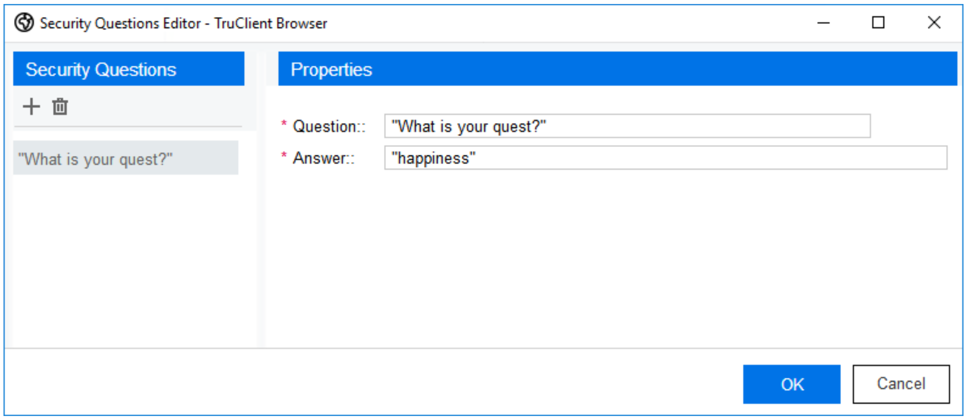
8. Click the **Step Editor** icon (›) to open the Step Editor for the step.
9. Click (expand) the **Security Questions** section.
10. Click + to open the Security Questions editor.
11. In the Security Questions pane of the Security Questions Editor, click + to add a new question.

    A new question appears with the default name "Question1." Its properties include the text box labeled Question (also shown with a default value of "Question1") and the text box labeled Answer, with a default value of "Answer1."

12. In the **Question** text box, type over the default text with the actual question exactly as it appears

on the login page, including capitalization and punctuation. The question in the left pane is simultaneously updated.

> **Important!** Be sure to enclose the text in quotation marks.

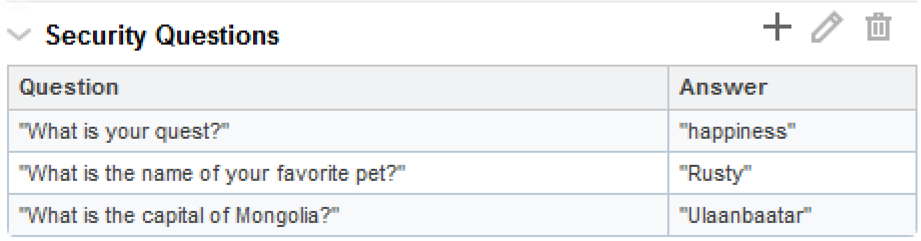


"What is your quest?"

* Question:: "What is your quest?"
* Answer:: "happiness"

13. In the **Answer** box, enter the correct response in quotation marks.
14. Repeat Steps 11 through 13 to add the information for the second question that might appear in the same location on the web page. In this example, the question is "What is the name of your favorite pet?"
15. Repeat Steps 11 through 13 to add the information for the third question that might appear in the same location on the web page. In this example, the question is "What is the capital of Mongolia?"
16. Click **OK**.

    The questions and answers are added to the table in the Security Questions section in the macro step.



Question	Answer
"What is your quest?"	"happiness"
"What is the name of your favorite pet?"	"Rusty"
"What is the capital of Mongolia?"	"Ulaanbaatar"

> **Tip:** If you later need to edit a question or answer, reopen the Security Questions Editor.

This completes the macro step for this particular location on the web page. To create more questions and answers for additional challenges, continue with "Adding Questions and Answers for Additional Challenges" on the next page.

Adding Questions and Answers for Additional Challenges

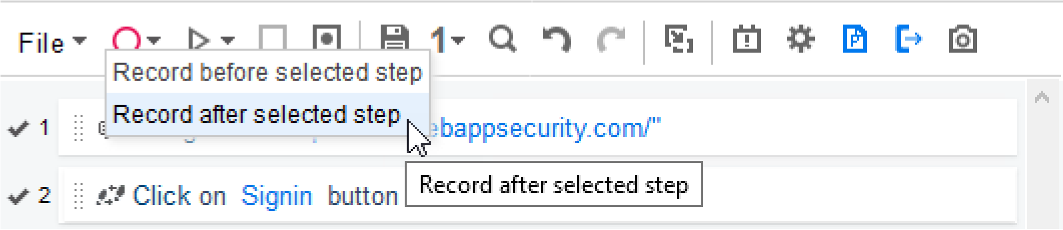
To add questions and answers for additional challenges:

1. Do one of the following to refresh the web page until the second set of questions appears:
   - Click in the Firefox browser window and press **F5**.
   - Right-click in the Firefox browser window and select the **Reload** icon.
2. Repeat Steps 2 through 16 of the preceding procedure to add another macro step for the second set of three questions and answers at the second location on the web page.
3. Do one of the following to refresh the web page until the third set of questions appears:
   - Click in the Firefox browser window and press **F5**.
   - Right-click in the Firefox browser window and select the **Reload** icon.
4. Repeat Steps 2 through 16 of "Recording a Macro for Challenge-Response Logins" on page 233 to add another macro step for the third set of three questions and answers at the third location on the web page.

Recording Additional Steps

If you need to record additional steps after creating steps for all possible question-and-answer combinations, then do the following:

1. In the TruClient sidebar, select the last step you created.
2. Click the drop-down arrow on the **Record** icon and select **Record after selected step**.


File
1
Record before selected step
Record after selected step
Record after selected step

✓ 1 ...ebappsecurity.com/"
✓ 2 Click on Signin button

3. Continue recording as usual.
4. Click the **Stop** icon (□).
5. Replay and save the macro.

Modifying the Macro Replay Level

As you record a macro, TruClient assigns a level from 1 to 3 to each step. For example, a level 1 step is essential to the macro. A click step that occurs in an area of the application that has no effect is assigned to level 2. Mouse-over steps are generally considered unnecessary for the macro and are assigned to level 3.

Macro steps are displayed and played with the granularity specified as level 1, 2, or 3 in the step level slider in the toolbar at the top of the TruClient browser. The highest granularity is level 3—setting the slider to level 3 displays and plays back all the steps at levels 1, 2, and 3. Using higher granularity might be required for successful playback, but it can cause the macro to take longer to run. By default, the Script Level is set to 1.

To modify a macro's replay level:

1. In the TruClient browser, click the step-level drop-down arrow (1▾).
2. Select the level of detail to view.

If you select a lower number, some steps are hidden. If you select a higher number, additional steps become visible.

Using the Event Handler Editor

If you record or add a step that does not occur in every run or occurs randomly, your macro might fail when the behavior does not occur. If there is no step to handle it, the macro might fail when the behavior does occur. An event handler calls a function to handle a condition only if the condition occurs.

Event Types

You can create an event handler for these types of events:

- **Object Event** - Used when an object-related event occurs. For example, a certain object appears on the page, or a certain object property gets a certain value.
- **Dialog Event** - Used when a dialog box pops up. For example, an alert or a prompt.

Creating an Event Handler

To create an event:

1. In the TruClient browser, click the **Event Handler Editor** icon (!).

   The Event Handler Editor appears.
2. Click + to add a new event handler.

   An error icon appears next to the handler name until enough information is added to define the handler.
3. Enter the properties for the event handler.

   For more information about event handlers and their properties, see the TruClient Help Center at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/global_event_handlers_c.htm.
4. Click **OK** to save the changes and close the editor.

Working with Logout Conditions

You can specify as many different logout conditions as you need, and if any one of them is met, Fortify WebInspect invokes the login macro to log back in and resume a scan where it left off. You can add, edit, and delete logout conditions using the Logout Condition Editor.

> **Important!** The final set of all logout conditions should cover all the cases of becoming logged out during a scan of the target site.

Accessing the Logout Condition Editor

To open the Logout Condition Editor:

- After recording a successful login, click the **Edit logout conditions** icon (↪).

  The Logout Condition Editor opens and displays all logout conditions already detected or created.

Adding a Logout Condition

To add a logout condition in the Logout Condition Editor:

1. Click ➕ in the left pane
2. In the Properties pane, type a name for the new condition in the **Name** field.

   The name in the left column is simultaneously updated with your changes.
3. Select which type of logout condition you want to use and complete the information required for that type. The following table describes the options.

Option	Description
Regex	With this option, you construct a regular expression (regex). A regular expression is a pattern that describes a set of strings. Regular expressions are constructed much like mathematical expressions by using various operators to combine smaller expressions. Only users with a working knowledge of regular expressions should use this feature. The regex must reflect the difference between a) the response to a logged-in user's request to access a protected page, and b) the response to the same request from the user, while *not* logged in, to access the same protected page. The general steps to construct the regex are as follows: a. Start the Web Proxy tool to record web traffic. For more information, see the Web Proxy Help or the *Tools Guide for Fortify WebInspect Products*.

Option	Description
	b. Log in to the target site and copy the URL of a protected page. c. Log out and use the copied URL to try to access the protected page without logging in. d. Compare the responses and identify a unique aspect of the response to the attempt to access the protected page without logging in. e. Open the Regular Expression Editor. For more information, see the Regular Expression Editor Help or the *Tools Guide for Fortify WebInspect Products*. f. Construct a regex that reflects the unique aspect of the response to the attempt to access the protected page without logging in. g. Copy the regex into the **Regex** field of the Logout Condition Editor.
URL	When you select this option, the currently displayed web page is automatically used as the default value. You can specify a static URL to which the target site redirects users when it logs them out. Do not specify the target site’s general login page.

4. Click **Close** to save the logout conditions and close the Logout Condition Editor.

Editing a Logout Condition

To edit an existing logout condition in the Logout Condition Editor:

1. Select the logout condition to edit in the left pane.

   The Properties pane lists the properties.
2. Edit the properties as needed.
3. Click **Close** to save the logout conditions and close the Logout Condition Editor.

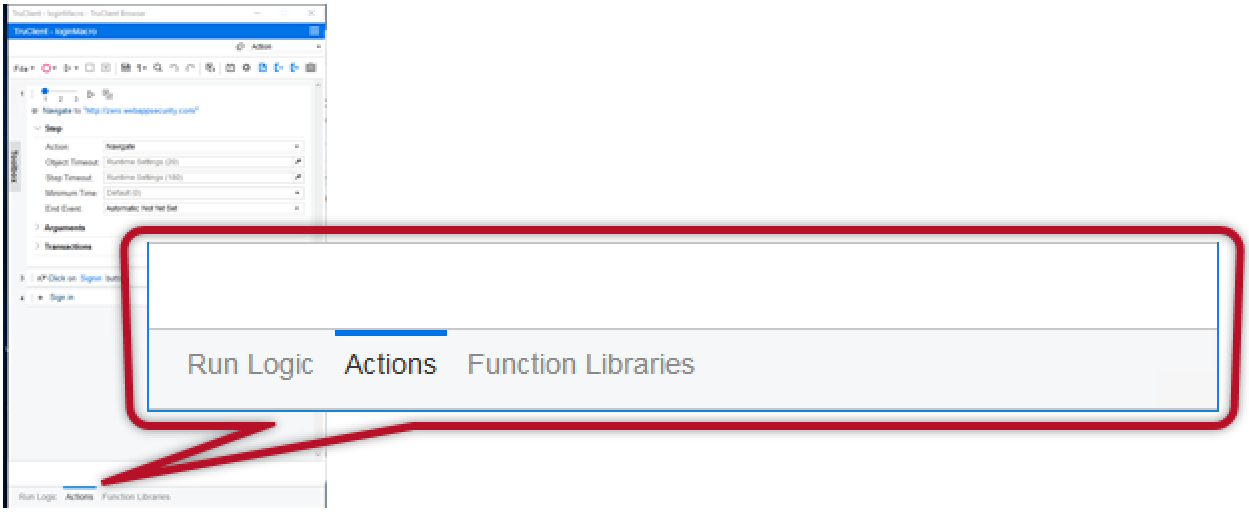
Deleting a Logout Condition

To delete an existing logout condition in the Logout Condition Editor:

1. Select the logout condition to delete in the left pane.
2. Click ✖.
3. Click **Close** to save the logout conditions and close the Logout Condition Editor.

Working with Actions

You create and run actions in the **Actions** tab, which you access from the bottom of the TruClient sidebar window.



TruClient - loginMacro
Action
File
Navigate to "http://zero.webappsecurity.com"
Step
Action: Navigate
Object Timeout: Runtime Settings (20)
Step Timeout: Runtime Settings (180)
Minimum Time: Default (0)
End Event: Automatic: Not Yet Set
Arguments
Transactions
Click on Signin button
Sign in
Run Logic
Actions
Function Libraries

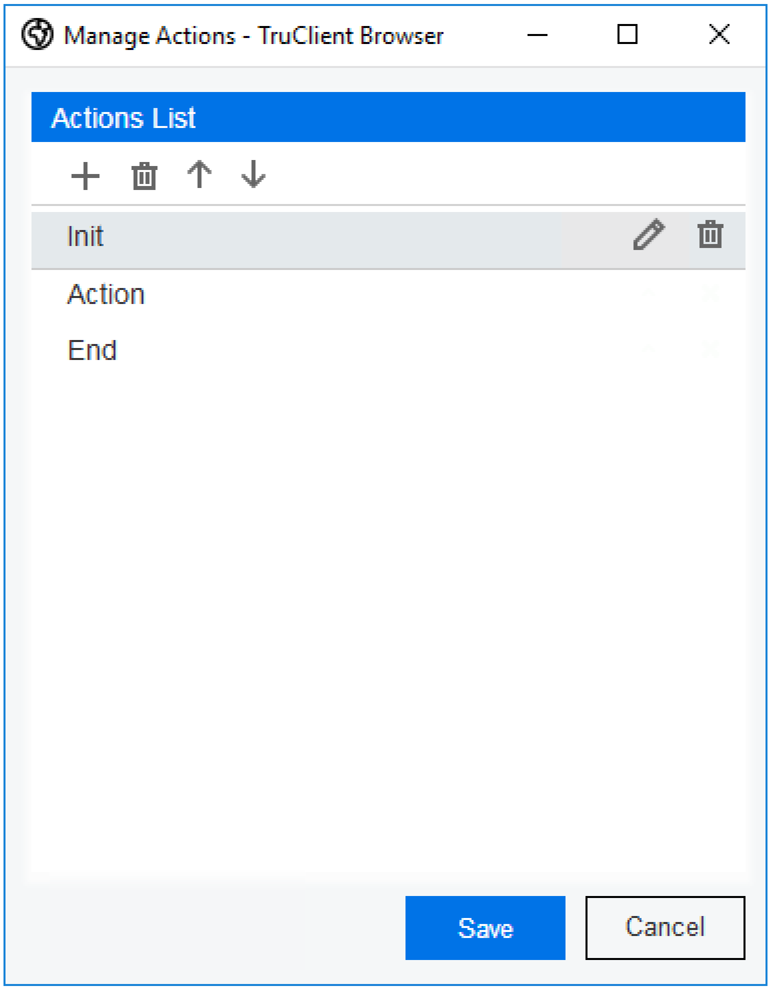
From this tab, you record, edit, and replay your macros.

Adding an Action to Your Macro

To add an action to your macro:

1. Click the **Manage Actions** icon (⚙) at the upper right hand corner of the TruClient sidebar window.

   The Manage Actions dialog box appears.



2. Click the **Add action** icon (+). Give the action a meaningful name.

Rearranging the Order of Actions

To rearrange the order of actions:

1. Click the **Manage Actions** icon ( ) at the upper right hand corner of the TruClient sidebar window.

   The Manage Actions dialog box appears.

2. Select an action.

3. Click the **Move up** or **Move down** icons (↑ ↓) to move the action up or down in the list.

Deleting an Action

To delete an action:

1. Click the **Manage Actions** icon (⚙) at the upper right hand corner of the TruClient sidebar window.

   The Manage Actions dialog box appears.
2. Select the action to delete.
3. Click the **Delete** icon (🗑).

Viewing Snapshots

TruClient generates snapshots during recording and replay. You can view snapshots in the TruClient snapshot viewer.

To view snapshots:

1. Click the **Snapshot view** icon (📷) in the TruClient sidebar window.

   The TruClient Snapshot Viewer opens to the right of the TruClient sidebar.
2. Choose the snapshot type to view as follows:
   - Click the **Recording** icon (○) to view the snapshots captured during recording.
   - Click the **Interactive Replay** icon (▷) to view the snapshots captured during interactive replay.
3. From the **View** drop-down list, select the viewing option as described in the following table.

Option	Description
Single	Displays the snapshot for a single step.
Compare	Splits the screen so that you can compare snapshots from different modes, such as snapshots captured during recording versus during interactive replay mode.
Thumbnail	Displays the snapshots in thumbnails view.

4. When viewing snapshots for a single step, navigate through the snapshots as follows:
   - Click the **Previous** or **Next** icons (‹ ›) to move to the snapshot for the previous or next step. The corresponding step is highlighted in the TruClient sidebar.
   - Click the **Play Slide Show** icon (▣) to automatically navigate through the snapshots in order.

     Click the **Stop Slide Show** icon (▣) to stop the slideshow.

Parameters Dialog

When recording a macro, you can use the Parameters Dialog to do the following:

- Create parameters for the user name and password to allow testers to use their own authentication credentials when starting a scan or to use multiple credentials for a multi-user login scan. For more information, see "Using Name and Password Parameters" below.
- Create a parameter for the URL to allow testers to designate an alternate URL when the macro runs. For example, suppose you record a macro for www.testsite.com. Later, you rename the site to www.testsite2.com. If you parameterize the URL when you record the macro, you do not need to record a new macro. You simply enter a new host name as the Start URL when you run a scan. For more information, see "Using a URL Parameter" on page 246.

Case-sensitive Parameter Names

Parameter names are case sensitive and must contain lowercase letters only.

Policy

The Policy settings that are visible in the Parameters Dialog are not applicable to Fortify WebInspect.

Using Name and Password Parameters

After creating and testing your login macro, you can create name and password parameters and assign them to recorded steps in the macro. The process involves creating parameters and replacing the recorded values with parameter names in the appropriate steps.

Creating Parameters

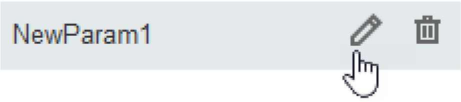
Use the Parameters Dialog to create name and password parameters for the macro.

To create name and password parameters:

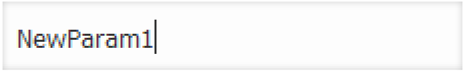
1. In the TruClient sidebar, click the **Edit Parameters** icon (🗎).

   The Parameter Dialog opens.
2. Click **Add Parameter**.

   A new parameter is created with a default name.
3. Click the edit icon for the newly created parameter.



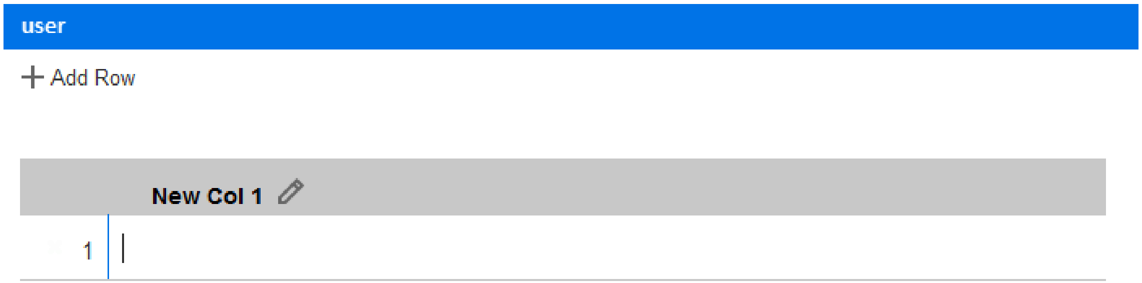
The parameter name becomes editable.



4. Replace the default name with a meaningful name for the parameter, such as `user`, and then press **Enter**.

   > **Important!** Parameter names are case sensitive and must contain lowercase letters only.

5. In the right-hand side of the window, place your cursor in the first row.

6. Type the first value to use during macro replay.

   > **Tip:** To edit the column name, click the edit icon in the column heading and type a new column name, such as `User Names`.

7. (Optional) To add another value (for example, to create a list of user names for a multi-user login scan):

   a. Click **Add Row**.

   b. Place your cursor in the new row.

   c. Type the next value to use during macro replay.

   d. Repeat Steps a through c for each additional value to add.

8. Click **Add Parameter**.

   A second parameter is created with a default name.

9. Click the edit icon for the newly created parameter.

10. Replace the default name with a meaningful name for the parameter, such as `pwd`, and then press **Enter**.

    > **Important!** Parameter names are case sensitive and must contain lowercase letters only.

11. In the right-hand side of the window, place your cursor in the first row.

12. Type the first value to use during macro replay.

13. (Optional) To add another value (for example, to create a list of passwords for a multi-user login scan):

    a. Click **Add Row**.

    b. Place your cursor in the new row.

   c. Type the next value to use during macro replay.
   d. Repeat Steps a through c for each additional value to add.

14. Click **OK** to save the parameters to the macro and close the Parameters Dialog.

Policy

The Policy settings that are visible in the Parameters Dialog are not applicable to Fortify WebInspect.

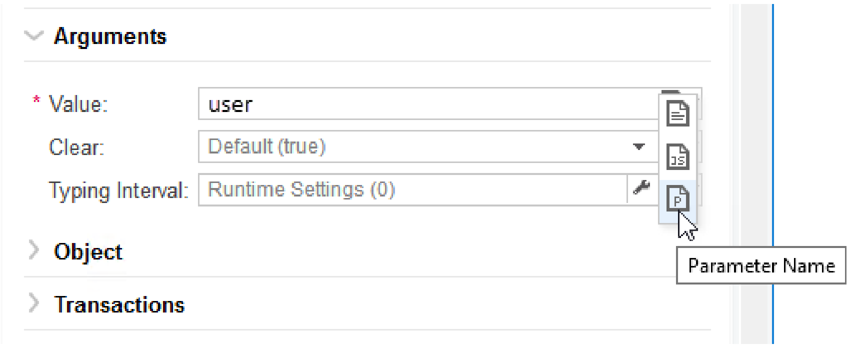
Assigning Parameters to Steps

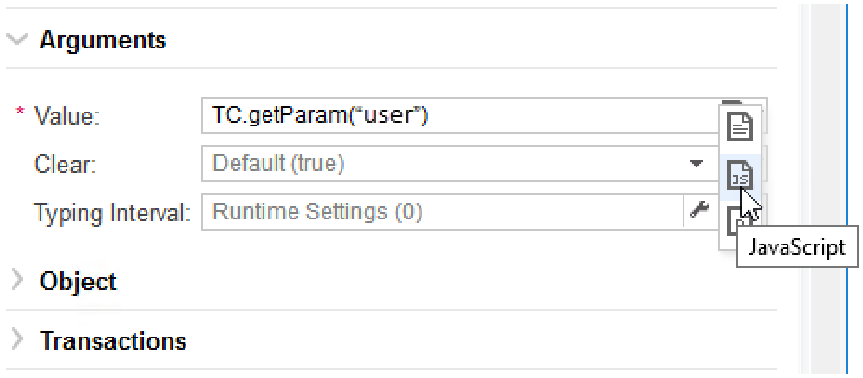
After creating the name and password parameters, use the Step Editor to add them to the appropriate steps.

1. In the TruClient sidebar, hover your mouse over the step that contains the user name.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor for the step.
3. Click (expand) **Arguments**.
4. Continue according to the following table.

To use...	Then...
Parameter Name	a. Highlight the entire contents of the **Value** text box, and type the variable name to use during playback, such as `user`. b. Select the **Parameter Name** icon from the drop-down list at the right of the Value text box.  Arguments * Value: user Clear: Default (true) Typing Interval: Runtime Settings (0) Object Transactions
JavaScript	a. Highlight the entire contents of the **Value** text box, and type the following: `TC.getParam(“<parameter name>”)` Substitute *`<parameter name>`* with the variable name to use during playback, such as `user`. b. Select the **JavaScript** icon from the drop-down list at the right of the Value text box.

To use...	Then...
	 Arguments * Value: TC.getParam("user") Clear: Default (true) Typing Interval: Runtime Settings (0) JavaScript Object Transactions

5. Hover your mouse over the step that contains the password.

   The Step Editor icon becomes visible.

6. Click the **Step Editor** icon (›) to open the Step Editor for the step.
7. Click (expand) **Arguments**.
8. Continue according to the following table.

To use...	Then...
Parameter Name	a. Highlight the entire contents of the **Value** text box, and type the variable name to use during playback, such as `pwd`. b. Select the **Parameter Name** icon from the drop-down list at the right of the Value text box.
JavaScript	a. Highlight the entire contents of the **Value** text box, and type the following: `TC.getParam(“<parameter name>”)` Substitute *`<parameter name>`* with the variable name to use during playback, such as `pwd`. b. Select the **JavaScript** icon from the drop-down list at the right of the Value text box.

9. Play the macro to verify that it logs in correctly.
10. Save the macro.

Using a URL Parameter

After creating and testing your login macro, you can create a URL parameter and assign it to a recorded step in the macro. The process involves creating a parameter and replacing the recorded URL with the parameter name in the appropriate step.

Creating the Parameter

Use the Parameters Dialog to create the URL parameter for the macro.

To create the URL parameter:

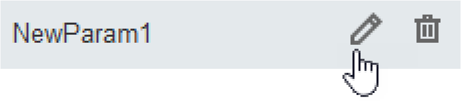
1. In the TruClient sidebar, click the **Edit Parameters** icon ().

   The Parameter Dialog opens.

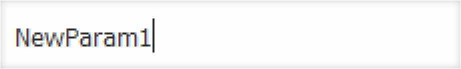
2. Click **Add Parameter**.

   A new parameter is created with a default name.

3. Click the edit icon for the newly created parameter.

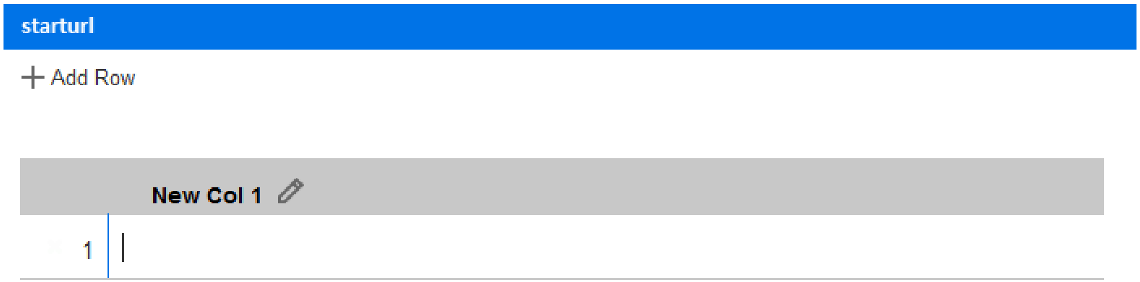
   The parameter name becomes editable.

4. Replace the default name with a meaningful name for the parameter, such as `starturl`, and then press **Enter**.

   Important! Parameter names are case sensitive and must contain lowercase letters only.

5. In the right-hand side of the window, place your cursor in the first row.

6. Type the URL to use during macro replay.

   Tip: To edit the column name, click the edit icon in the column heading and type a new column name, such as `URLs List`.

7. Click **OK** to save the parameter to the macro and close the Parameters Dialog.

Policy

The Policy settings that are visible in the Parameters Dialog are not applicable to Fortify WebInspect.

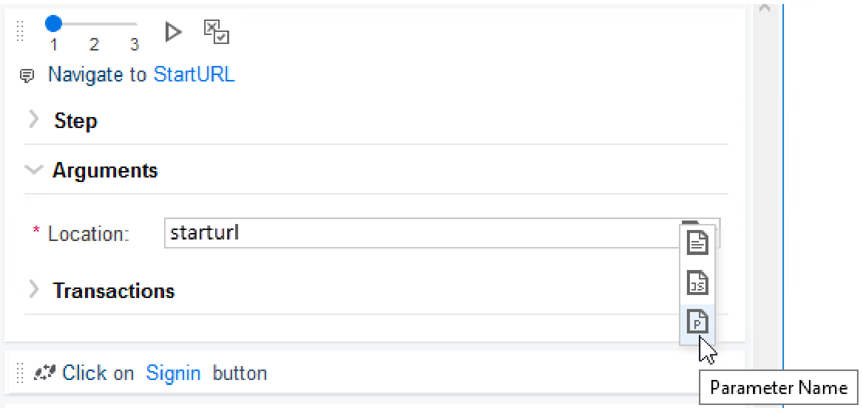
Assigning the Parameter to a Step

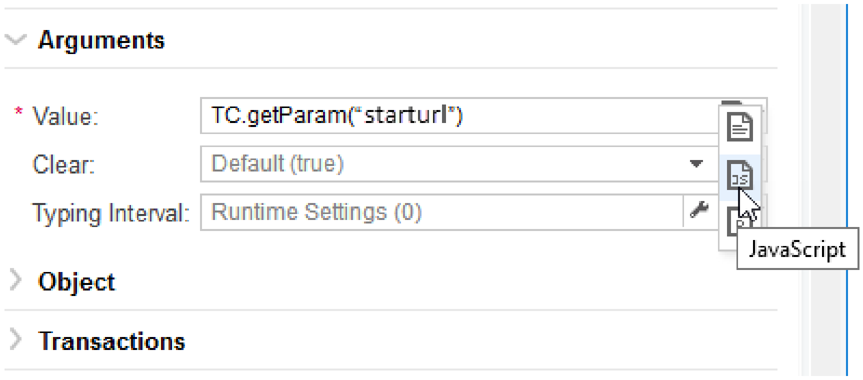
After creating the URL parameter, use the Step Editor to add it to the appropriate step.

1. In the TruClient sidebar, hover your mouse over the step that contains the URL ("Navigate to...").

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor for the step.
3. Click (expand) **Arguments**.
4. Continue according to the following table.

To use...	Then...
Parameter Name	a. Highlight the entire contents of the **Location** text box (including quotation marks), and type the variable name to use during playback, such as `starturl`. b. Select the **Parameter Name** icon from the drop-down list at the right of the Value text box.  1 2 3 Navigate to StartURL Step Arguments * Location: starturl Transactions Click on Signin button Parameter Name
JavaScript	a. Highlight the entire contents of the **Value** text box, and type the following: `TC.getParam(“<parameter name>”)` Substitute *`<parameter name>`* with the variable name to use during playback, such as `starturl`. b. Select the **JavaScript** icon from the drop-down list at the right of the Value text box.

To use...	Then...
	 Arguments * Value: TC.getParam("starturl") Clear: Default (true) Typing Interval: Runtime Settings (0) Object Transactions

5. Play the macro to verify that it accesses the start URL.
6. Save the macro.

Step Arguments not Related to Objects

The following tables describe the step arguments not related to objects. The actions in these step arguments do not operate on objects. Therefore, they do not have a role.

> **Tip:** Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Evaluate JavaScript

The Evaluate JavaScript action runs the JavaScript code contained in the step. The following table describes the step argument for the Evaluate JavaScript action.

Argument	Description
Code	Specifies the JavaScript code to run.

Evaluate C

The Evaluate C action receives a C function name and runs it. You can define functions in the `C-functions.c` file.The following table describes the step arguments for the Evaluate C action.

Argument	Description
Function Name	Specifies the name of the C function to run.
Function Timeout	Defines the amount of time to wait for the function to run (in seconds),

Argument	Description
	before it returns an error.

Call Function

Enter values for arguments that are set by the called function (for example, Arg1, Arg2).

Catch Error

The Catch Error action catches an error in the step immediately preceding and runs the contents of the catch error step. The following table describes the step argument for the Evaluate C action.

Argument	Description
Error Type.	Specifies the error type you want to catch: • **Any** • **Object identification** - Indicates that the object the action is performed on cannot be found. • **Step arguments** - Indicates that one or more of the arguments to the preceding step is invalid. For example, the data type is wrong. • **Step Action** - Indicates that the user action failed. For example, a navigation step did not find the page. For an action on a UI element, this error is triggered if the object was found and the action failed anyway.

For Loop

The For Loop is a logical structure that repeats the steps contained in the loop a specified number of times.The following table describes the step arguments for the For Loop action.

Argument	Description
Init	Specifies the condition for the initialization operation, which must be met before testing the condition of the first iteration.
Condition	Specifies the condition for continuing to the next iteration. Options are: • **true** - Indicates that the specified condition is met. • **false** - Indicates that the specified condition is not met. • **Regular expression** - Defines a regular expression as the condition.

Argument	Description
Increment	Increments a counter in the condition.

Generic API Action

The Generic API Action are blank steps that can be inserted and manually configured. The arguments vary according to the API selected. For more information, see the TruClient API help at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/tocs/_tc_toc_functionReference.htm.

The following table describes the step argument for the Generic API Action.

Argument	Description
Variable	Specifies the name of the JavaScript variable in which the returned value is stored.

If Block

The If Block action is a logical structure that runs the steps contained in the block if the condition is met. The following table describes the step argument for the If Block action.

Argument	Description
Condition	Specifies the condition for continuing to the next iteration. Options are: • **true** - Indicates that the specified condition is met (this is the default setting). • **false** - Indicates that the specified condition is not met. • **Regular expression** - Defines a regular expression as the condition.

Rendezvous

The Rendezvous action synchronizes all Vusers in a Controller scenario to run a specified step at the same time. The following table describes the step argument for the Rendezvous action.

Argument	Description
Rendezvous Name	Identifies the name for the rendezvous, which is the specific point where Vusers are synchronized in the step.

Verify PDF Content

The Verify PDF Content action verifies the content of a PDF document during script execution. The following table describes the step arguments for the Verify PDF Content action.

Argument	Description
Path	Specifies the file location. The location can be either a local file or a Web page address.
Value	Specifies the PDF string to verify. The value can be either plain text or JavaScript code.
Condition	Specifies how to verify the value. Options are: **Contain** - Indicates that the PDF contains the specified value. **Not Contain** - Indicates that the PDF does not contain the specified value. **Regular expression** - Defines a regular expression to search the PDF content.

The following are known limitations for the Verify PDF Content action:

- You cannot search for content across a page break.
- You can cannot search for a value that contains both special, such as Chinese or Japanese characters, and alphanumeric characters.

Wait

The Wait action waits for a specified number of seconds (or milliseconds), before continuing with the next step. The following table describes the step arguments for the Verify PDF Content action.

Argument	Description
Interval	Specifies the time value that the step will wait for, before the step passes. The default value is 3.
Unit	Specifies the interval value. The unit properties available are Seconds (this is the default setting) and Milliseconds.
Think Time	Specifies whether to include the wait time in the think time calculation. The default setting is true.

Audio Role

The following table describes the step argument for the **Seek** action of the audio role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Time	Sets or returns the current position (in seconds) of the audio playback.

Browser Role

The following tables describe the step arguments related to browser role objects.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Activate

The following table describes the step arguments for the Activate action.

Argument	Description
Ordinal	Defined as an integer.
Title	Defined as a string. **Note:** The title is automatically updated during recording and can be set as an alternative step.

Activate Tab

The following table describes the step arguments for the Activate Tab action.

Argument	Description
Ordinal	Specifies which tab (integer) to activate.
Title	Defined as a string. **Note:** The title is automatically updated during recording and can be set as an alternative step.

Close Tab

The following table describes the step arguments for the Close Tab action.

Argument	Description
Ordinal	Specifies which tab (integer) to close.
Title	Moves the specified browser window to the foreground. Defined as a string. **Note:** The title is automatically updated during recording and can be set as an alternative step.

Add Tab

The following table describes the step arguments for the Add Tab action.

Argument	Description
Location	Specifies the URL to navigate to in the newly opened tab.
Window	Points to the global window object of the application. **Note:** The `window.location` object cannot be used with Internet Explorer. Use the `document.URL` object instead.

Navigate

The following table describes the step argument for the Navigate action.

Argument	Description
Location.	Specifies the URL to navigate to.

Go Back

The following table describes the step argument for the Go Back action.

Argument	Description
Count	Specifies the number of pages to go back.

Go Forward

The following table describes the step argument for the Go Forward action.

Argument	Description
Count	Specifies the number of pages to go forward.

Resize

The following table describes the step arguments for the Resize action.

Argument	Description
Width	Specifies the new width. Leaving this blank means do not resize the width.
Height	Specifies the new height. Leaving this blank means do not resize the height.

Scroll

The following table describes the step arguments for the Scroll action.

Argument	Description
X Coordinate	Indicates the new x coordinate. Leaving this blank means do not scroll along the x axis.
Y Coordinate	Indicates the new y coordinate. Leaving this blank means do not scroll along the y axis.

Dialog - Confirm

The following table describes the step argument for the Dialog - Confirm action.

Argument	Description
Button	Indicates OK or Cancel.

Dialog Prompt

The following table describes the step arguments for the Dialog Prompt action.

Argument	Description
Value	Indicates the string to enter.
Button	Indicates OK or Cancel.

Dialog - Authenticate

The following table describes the step arguments for the Dialog - Authenticate action.

Argument	Description
Username	Specifies the username to enter.
Password	Specifies the password to enter.
Domain	Specifies the domain to enter.
Button	Indicates OK or Cancel.

Dialog - Prompt Password

The following table describes the step arguments for the Dialog - Prompt Password action.

Argument	Description
Password	Specifies the password to enter.
Button	Indicates OK or Cancel.

Verify

The following table describes the step arguments for the Verify action.

Argument	Description
Value	Indicates the value of the property to verify.
Property	Identifies the property to verify. You can verify the following properties of a browser object: • **Title** - Specifies the title of the browser window.

Argument	Description
	• **Location** - Specifies the location of the browser window.
Condition	Specifies the relationship between the value and property arguments.

Checkbox Role

The following table describes the step argument for the **Set** action of the checkbox role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Checked	Sets the check box to either checked (true) or unchecked (false).

Datepicker Role

The following table describes the step argument for the **Set Day** action of the datepicker role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Day	Represents the day of the month. Value is an integer between 1-31.

Element Role

The following tables describe the step arguments related to element role objects.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Mouse Actions

The following table describes the step arguments for the Mouse Down, Mouse Up, Mouse Over, Click, and Double Click mouse actions.

Note: Mouse Over does not have the X/Y Coordinate arguments.

Argument	Description
Button	Identifies the mouse button that is clicked.
X Coordinate	Identifies the offset location of the action relative to the upper left corner of the object. If not specified, the default is the center of the object.
Y Coordinate	Identifies the offset location of the action relative to the upper left corner of the object. If not specified, the default is the center of the object.
Ctrl Key	Indicates whether this key is pressed during the action.
Alt Key	Indicates whether this key is pressed during the action.
Shift Key	Indicates whether this key is pressed during the action.

Drag

The following table describes the step arguments for the Drag action.

Argument	Description
Button	Identifies the mouse button that is clicked.
X Offset	Indicates the amount of pixels to drag the object on the x axis. A positive number indicates a drag to the right.
Y Offset	Indicates the amount of pixels to drag the object on the y axis. A positive number indicates a drag down.
Path	Identifies the list of coordinates representing the user drag path. Do not modify this argument.
Ctrl Key	Indicates whether this key is pressed during the action.
Alt Key	Indicates whether this key is pressed during the action.
Shift Key	Indicates whether this key is pressed during the action.

Note: The **X Offset**, **Y Offset**, and **Path** arguments are mutually exclusive.

Drag To

The following table describes the step arguments for the Drag To action.

Argument	Description
Target Object	Indicates that the step object is dragged to this target object.
HTML 5	Provides drag and drop support to the browser making it easier to code. When this argument is "true", only the "Target Object" and "HTML5" arguments are visible. When it is "false", the other arguments are also visible.
Button	Identifies the mouse button that is clicked.
X Offset	Identifies the offset from the top left of the target object in the x axis. This number must be positive.
Y Offset	Identifies the offset from the top left of the target object in the y axis. This number must be positive.
Ctrl Key	Indicates whether this key is pressed during the action.
Alt Key	Indicates whether this key is pressed during the action.
Shift Key	Indicates whether this key is pressed during the action.

Get Property

The following table describes the step arguments for the Get Property action.

Argument	Description
Property	Indicates the property whose value will be stored in the specified variable. The list of properties available depends on all the roles of the object. The following are the default properties available for all objects: • **Visible text** - Indicates the visible text of the item, corresponding to the DOM textContent property. • **All text** - Indicates the entire text of the item, corresponding to the DOM textContent property. • **Inner HTML** - Indicates the inner html markup of the object, corresponding to the DOM innerHTML property.

Argument	Description
Variable	Indicates the name of the variable in which to store the specified property value.

Scroll

The following table describes the step arguments for the Scroll action.

Argument	Description
Horizontally	Specifies the distance (in pixels) to scroll horizontally.
Vertically	Specifies the distance (in pixels) to scroll vertically.

Note: Both arguments must be integers, with a minimum and default value of 0. The scrolling is done on the containing document rather than on the element itself.

Verify

The following table describes the step arguments for the Verify action.

Argument	Description
Value	Indicates the string or number to verify.
Property	Indicates the object property whose value will be verified. The list of properties available to verify depends on all the roles of the object. The following are the default properties available for verification on all objects: • **Visible text** - Identifies items that are visible in the application. • **All text** - Identifies items that are in the application but are not necessarily visible. Items in this category are contained in DOM property textContent. • **Inner HTML** - Identifies items that are contained in the DOM property innerHTML.
Condition	Indicates the relationship between the value and property arguments.

Wait for Property

The following table describes the step arguments for the Wait for Property action.

Argument	Description
Value	Indicates the value of the specified property that the step will wait for, before the step passes.
Property	Indicates the object property whose value the script will wait for. The list of properties available for which to wait, depends on all the roles of the object. The following are the default properties available for all objects: • **Visible text** - Identifies items that are visible in the application. • **All text** - Identifies items that are in the application but are not necessarily visible. Items in this category are contained in DOM property textContent. • **Inner HTML** - Identifies items that are contained in the DOM property innerHTML.
Condition	Indicates the relationship between the value and property arguments.

Filebox Role

The following table describes the step argument for the **Set** action of the filebox role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Path	Specifies the selected path.

Flash Object Role

The following table describes the step argument for the **Type** action of the flash object role.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Value	Specifies what is typed.

Focusable Role

The following table describes the step arguments for the **Press Key** action of the focusable role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Key Name	Specifies Enter or Space.
Ctrl Key	Indicates whether this key is pressed during the action.
Alt Key	Indicates whether this key is pressed during the action.
Shift Key	Indicates whether this key is pressed during the action.

Listbox Role

The following table describes the step arguments for the **Select** action of the listbox role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Text	Indicates the selected string or a regular expression. This value is optional.
Ordinal	Specifies the order of the selected item in the list. If the text argument is also specified, than this argument refers to the instance of the specified text value in the listbox. An ordinal of 0 generates a random value. If both text and ordinal are left empty, then the default ordinal (1) is automatically filled.
Inner Object	Allows selecting an option based on TruClient's object identification mechanism for the option element itself, rather than identifying its container object and specifying an Ordinal.

Multi_listbox Role

The following tables describe the step arguments related to multi_listbox role objects.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Select

The following table describes the step arguments for the Select action.

Argument	Description
Text	Indicates the selected string or a regular expression.
Ordinal	Specifies the order of the selected item in the list. If the text argument is also specified, then this argument refers to the instance of the specified text value in the listbox. An ordinal of 0 generates a random value.

Multi Select

The following table describes the step arguments for the Multi Select action.

Argument	Description
Text	The option's text.
By Ordinal	Specifies the ordinals of the item's Delimiter.
Delimiter	Specifies the characters used to separate the selected values.

Radiogroup Role

The following table describes the step arguments for the **Select** action of the radiogroup role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Text	Indicates the selected string or regular expression.
Ordinal	Specifies the order of the selected item in the list. If the text argument is also specified, than this argument refers to the instance of the specified text value in the listbox. An ordinal of 0 generates a random value.

Slider Role

The following table describes the step argument for the **Set** action of the slider role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Value	Specifies the value to which the slider is set.

Textbox Role

The following table describes the step arguments for the **Type** action of the textbox role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Value	Indicates what is typed.
Clear	Clears the text box before typing. The default is true.
Typing Interval	Indicates the average time in milliseconds between keystrokes.

Video Role

The following table describes the step argument for the **Seek** action of the video role object.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Argument	Description
Time	Sets or returns the current position (in seconds) of the video playback.

Step Arguments not Related to Objects

The following tables describe the step arguments not related to objects. The actions in these step arguments do not operate on objects. Therefore, they do not have a role.

Tip: Mandatory arguments are marked with a red star to the left of the argument name in the user interface. All arguments can accept JavaScript code and TruClient functions as values.

Evaluate JavaScript

The Evaluate JavaScript action runs the JavaScript code contained in the step. The following table describes the step argument for the Evaluate JavaScript action.

Argument	Description
Code	Specifies the JavaScript code to run.

Evaluate C

The Evaluate C action receives a C function name and runs it. You can define functions in the `C-functions.c` file.The following table describes the step arguments for the Evaluate C action.

Argument	Description
Function Name	Specifies the name of the C function to run.
Function Timeout	Defines the amount of time to wait for the function to run (in seconds), before it returns an error.

Call Function

Enter values for arguments that are set by the called function (for example, Arg1, Arg2).

Catch Error

The Catch Error action catches an error in the step immediately preceding and runs the contents of the catch error step. The following table describes the step argument for the Evaluate C action.

Argument	Description
Error Type.	Specifies the error type you want to catch: • **Any** • **Object identification** - Indicates that the object the action is performed on cannot be found. • **Step arguments** - Indicates that one or more of the arguments to the preceding step is invalid. For example, the data type is wrong.

Argument	Description
	• **Step Action** - Indicates that the user action failed. For example, a navigation step did not find the page. For an action on a UI element, this error is triggered if the object was found and the action failed anyway.

For Loop

The For Loop is a logical structure that repeats the steps contained in the loop a specified number of times.The following table describes the step arguments for the For Loop action.

Argument	Description
Init	Specifies the condition for the initialization operation, which must be met before testing the condition of the first iteration.
Condition	Specifies the condition for continuing to the next iteration. Options are: • **true** - Indicates that the specified condition is met. • **false** - Indicates that the specified condition is not met. • **Regular expression** - Defines a regular expression as the condition.
Increment	Increments a counter in the condition.

Generic API Action

The Generic API Action are blank steps that can be inserted and manually configured. The arguments vary according to the API selected. For more information, see the TruClient API help at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/tocs/_tc_toc_functionReference.htm.

The following table describes the step argument for the Generic API Action.

Argument	Description
Variable	Specifies the name of the JavaScript variable in which the returned value is stored.

If Block

The If Block action is a logical structure that runs the steps contained in the block if the condition is met. The following table describes the step argument for the If Block action.

Argument	Description
Condition	Specifies the condition for continuing to the next iteration. Options are: • **true** - Indicates that the specified condition is met (this is the default setting). • **false** - Indicates that the specified condition is not met. • **Regular expression** - Defines a regular expression as the condition.

Rendezvous

The Rendezvous action synchronizes all Vusers in a Controller scenario to run a specified step at the same time. The following table describes the step argument for the Rendezvous action.

Argument	Description
Rendezvous Name	Identifies the name for the rendezvous, which is the specific point where Vusers are synchronized in the step.

Verify PDF Content

The Verify PDF Content action verifies the content of a PDF document during script execution. The following table describes the step arguments for the Verify PDF Content action.

Argument	Description
Path	Specifies the file location. The location can be either a local file or a Web page address.
Value	Specifies the PDF string to verify. The value can be either plain text or JavaScript code.
Condition	Specifies how to verify the value. Options are: **Contain** - Indicates that the PDF contains the specified value. **Not Contain** - Indicates that the PDF does not contain the specified value.

Argument	Description
	Regular expression - Defines a regular expression to search the PDF content.

The following are known limitations for the Verify PDF Content action:

- You cannot search for content across a page break.
- You can cannot search for a value that contains both special, such as Chinese or Japanese characters, and alphanumeric characters.

Wait

The Wait action waits for a specified number of seconds (or milliseconds), before continuing with the next step. The following table describes the step arguments for the Verify PDF Content action.

Argument	Description
Interval	Specifies the time value that the step will wait for, before the step passes. The default value is 3.
Unit	Specifies the interval value. The unit properties available are Seconds (this is the default setting) and Milliseconds.
Think Time	Specifies whether to include the wait time in the think time calculation. The default setting is true.

Enhancing Macros

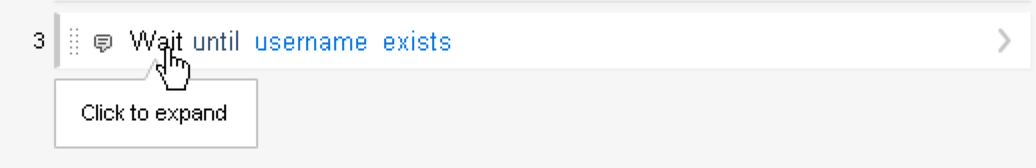
You can incorporate the following optional enhancements to recorded macros:

- "Modifying Steps" below
- "Inserting Loops and Loop Modifiers" on the next page
- "Inserting If Blocks, If-else Blocks, and Exit Steps" on page 270
- "Inserting Comments " on page 271
- "Inserting Catch Error Steps" on page 271
- "Verifying that an Object Exists" on page 272
- "Inserting Generic Steps" on page 272

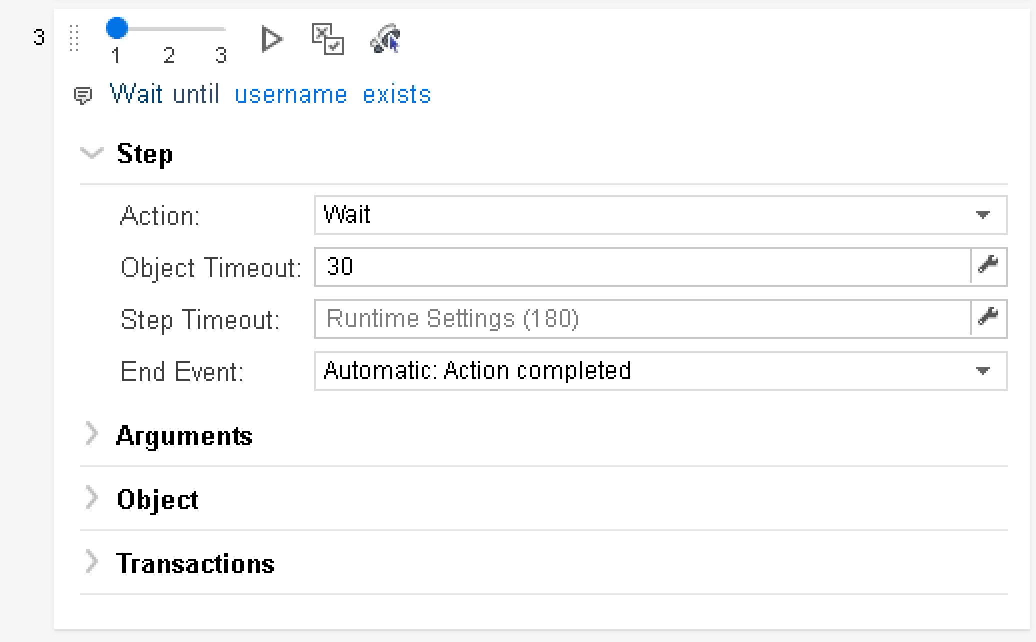
Modifying Steps

To modify step arguments and objects:

- Select the desired step and expand the options.



This expands the step and allows you to modify the objects and properties.



Action: Wait

Object Timeout: 30

Step Timeout: Runtime Settings (180)

End Event: Automatic: Action completed

For details about the available options, see "Understanding the Toolbox" on page 227.

Inserting Loops and Loop Modifiers

Loops repeat selected portions of the macro until certain criteria are met or for a specified number of iterations. You can insert loops and break/continue loop modifiers from the Flow Control section of the Toolbox.

Inserting "For" Loops

"For" loops perform the steps surrounded by the loop until the end condition is met or the code reaches a break statement. Loop arguments use JavaScript syntax.

To insert a For loop:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Flow Control**.
3. Click and drag the **For loop** element to the desired location in the recorded steps.

Inserting "Break" Statements

Break statements indicate that the current loop should end immediately. For example, if a Break statement is encountered in the second of five iterations in a For loop, the loop will end immediately without completing the remaining iterations.

To insert a Break statement:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Flow Control**.
3. Click and drag the **Break** element to the desired location in the recorded steps.

Inserting "Continue" Statements

Continue statements indicate that the current loop iteration should end immediately. The loop condition is then checked to see if the entire loop should end as well. For example, if a Continue statement is encountered in the second of five iterations in a For loop, the second iteration will end immediately and the third iteration will begin.

To insert a Continue statement:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Flow Control**.
3. Click and drag the **Continue** element to the desired location in the recorded steps.

Inserting If Blocks, If-else Blocks, and Exit Steps

To conditionalize a portion of the macro, you can insert If or If-else blocks. Exit steps cause a macro to exit the iteration or the entire macro. These can be used with If statements to exit a macro or iteration when a specified condition occurs.

For information about specific actions and arguments for each of these, see "Step Arguments not Related to Objects" on page 264.

Inserting an If Block

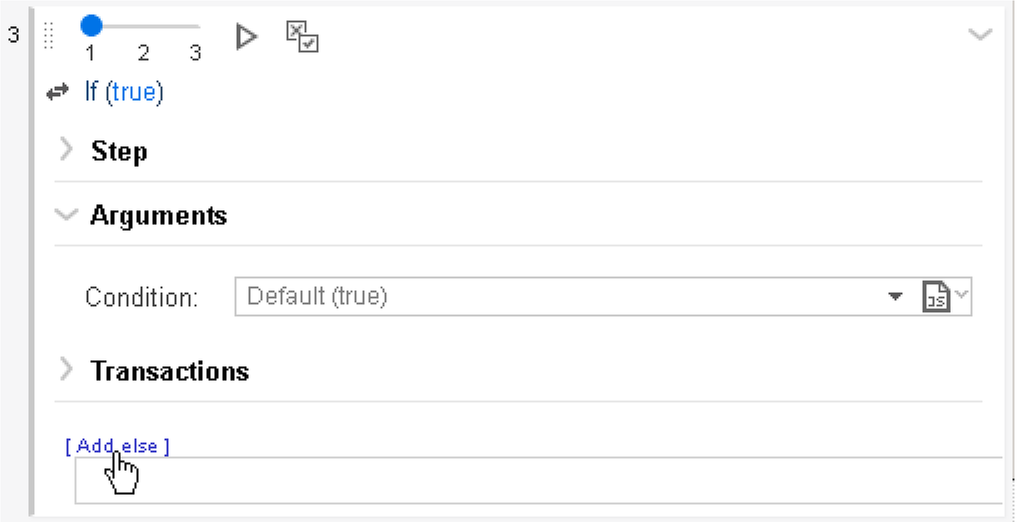
To insert an If block:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Flow Control**.
3. Click and drag the **If block** element to the desired location in the recorded steps.

Adding an Else Condition

To add an else condition:

1. Click the **Add else** link in the expanded step.



Condition: Default (true)

2. In the **Else** field, type the else condition.

Inserting an Exit Step

To insert an exit step:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Flow Control**.
3. Click and drag the **Exit** element to the desired location in the recorded steps.

Inserting Comments

You can add comments to your macro so that others can understand what specific steps in the macro accomplish.

To insert comments into your macro:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Miscellaneous**.
3. Click and drag the **Comment** element to the desired location in the recorded steps.
4. Type the comment in the space provided.

Inserting Catch Error Steps

"Catch error" steps are group steps that run their contents if the previous step contains an error. Additionally, the error is "caught" and is not returned. You can define catch error steps to catch any error, or a specific type of error. If there are two catch error steps in a row, they both apply to the same step.

> **Tip:** To group steps, use **Ctrl** + click to select multiple steps, right-click any of them, and click **Group Steps**.

To insert a catch error step:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Flow Control**.
3. Click and drag the **Catch Error** element to the desired location in the recorded steps.
4. Expand the catch error step and configure the argument. For more information, see "Step Arguments not Related to Objects" on page 264.

Verifying that an Object Exists

You can insert a verify step to verify that a string or object exists in the application.

To insert a verify step:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Functions**.
3. Click and drag the **Verify** element to the desired location in the recorded steps.
4. In the verify step, click the **Click to choose an object** link.
5. In the Firefox browser, select the object you want to verify.

Inserting Generic Steps

You can insert a blank or generic step and manually configure it.

To insert a generic step:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Functions**.
3. Click and drag the **Generic Object Action** element or the **Generic Browser Action** element to the desired location among the macro steps.

   > **Tip:** Generic Object Actions perform an unspecified action on an object. Generic Browser Actions perform an unspecified action on the browser such as go back, reload, switch tabs, and so on.

4. Expand the step, and configure the step properties. For more information, see "Step Arguments not Related to Objects" on page 264.

Debugging Macros

You can try these tasks to interactively debug a macro:

- "Viewing Replay Errors" below
- "Running the Macro Step by Step" below
- "Using Breakpoints" on the next page
- "Modifying Step Levels" on the next page
- "Inserting a Wait Step" on page 275
- "Disabling/Enabling Steps" on page 275
- "Making a Step Optional" on page 276
- "Playing a Step" on page 276
- "Playing From a Step to End of Macro" on page 276

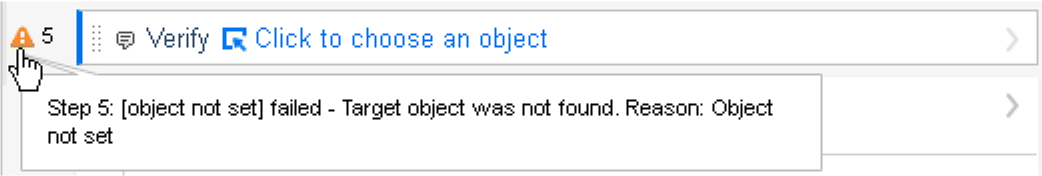
Viewing Replay Errors

If any steps failed during replay, they are marked with an error icon (⚠).

To view details about the error:

- Hover the mouse pointer over the error icon.

  A description of the error appears.



Step 5: [object not set] failed - Target object was not found. Reason: Object not set

Running the Macro Step by Step

The step-by-step replay pauses the replay after each step, which allows you to view the sequence more slowly and in a controlled manner.

To run the macro step by step:

- In the TruClient sidebar, select the down arrow next to the **Replay** icon (▷ ▾) and select **Replay step by step**.

  The first (or next) step plays and the replay stops.

Repeat this procedure after each step to continue the step-by-step replay.

Using Breakpoints

Breakpoints instruct the macro to stop running during a replay. You can insert (or toggle on) breakpoints to help debug a macro. After inserting a breakpoint on a step, the macro plays to the breakpoint and pauses. At this point, the Inspector Panel opens at the bottom of the TruClient browser. You can then continue playing the macro from the breakpoint.

> **Note:** The TruClient Web Macro Recorder adds a breakpoint automatically if the macro fails during playback.

Inserting a Breakpoint

To insert a breakpoint:

1. In the TruClient browser, select the step where you want to insert the breakpoint.
2. Click the toggle breakpoint icon (▣).

A breakpoint is added to the step.

Deleting a Breakpoint

To delete a breakpoint:

1. In the TruClient browser, select the step where the breakpoint has been inserted.
2. Click the toggle breakpoint icon (▣).

The breakpoint is removed from the step.

Modifying Step Levels

As you record a macro, TruClient assigns a level from 1 to 3 to each step. For example, a level 1 step is essential to the macro. A click step that occurs in an area of the application that has no effect is assigned to level 2. Mouse-over steps are generally considered unnecessary for the macro and are assigned to level 3.

Macro steps are displayed and played with the granularity specified as level 1, 2, or 3 in the step level slider in the toolbar at the top of the TruClient browser. The highest granularity is level 3—setting the slider to level 3 displays and plays back all the steps at levels 1, 2, and 3. Using higher granularity might be required for successful playback, but it can cause the macro to take longer to run. By default, the Script Level is set to 1.

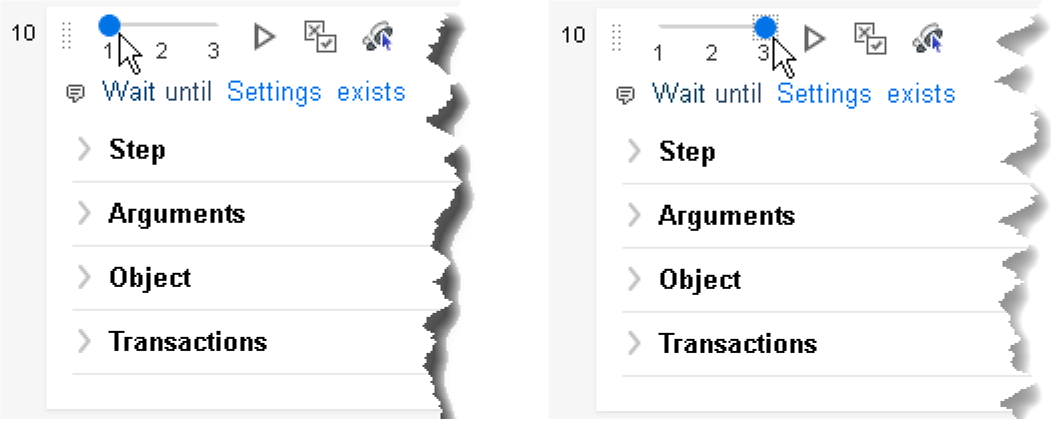
In certain cases, you may want to manually change the level of a particular step, not the entire macro. For example, you may want to display and play a particular mouse-over step.

To change the level of a step:

1. In the TruClient sidebar, hover your mouse over the step to change.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor.
3. Move the slider at the top of the Step Editor (to the right of the step number) to the desired level.



Important! If the step is part of a group step, both the group step and the individual step must be modified.

Tip: To group steps, use **Ctrl** + click to select multiple steps, right-click any of them, and click **Group Steps**.

See Also

"Modifying the Macro Replay Level" on page 236

Inserting a Wait Step

Wait steps cause the macro to pause for a specified amount of time before continuing with the next step. Wait for Object steps cause the macro to wait for a specified object to appear in the application before continuing with the next step. Wait steps begin after the End Event of the previous step is reached. This means that the previous step may continue to run after the wait step has been reached.

To insert a wait step:

1. Click the **Toolbox** vertical tab on the left side of the TruClient sidebar.
2. Click (expand) **Functions**.
3. Click and drag the **Wait** element or the **Wait for Object** element to the location you want in the recorded steps.
4. If you inserted a Wait for Object step, select the **Click to choose an object** link to select the target object in the application.

Disabling/Enabling Steps

You can disable recorded steps so that they remain in the macro and can be re-enabled in the future, but are not played.

To disable/enable a macro step during replay:

1. In the TruClient sidebar, hover your mouse over the step to change.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor.
3. Click the **Disable/Enable during replay** icon (☒) in the toolbar for the step.

Tip: Alternatively, to disable or re-enable one or more steps, use Ctrl + click to select them, right-click one of the steps, and click **Disable Steps** or **Enable Steps** on the popup menu.

Making a Step Optional

You can make some steps optional. An optional step is skipped during replay if its object is not found.

To make a step optional:

1. In the TruClient sidebar, hover your mouse over the step to change.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor.
3. Click the **Set step as optional** icon ( ) in the step toolbar.

Tip: To make a step non-optional again, click the icon again.

Playing a Step

You can play a specific step to inspect the activity recorded in the step.

To play one step:

1. In the TruClient sidebar, hover your mouse over the step.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor.
3. Click the **Play this step only** icon (▷) in the step toolbar.

Playing From a Step to End of Macro

To start playback at a particular step and continue until the end of the macro:

1. Select the starting step.
2. Right-click the step, and then select **Play From This Step** on the context menu.

Resolving Object Identification Issues

In dynamic websites, objects that have been recorded can often move or change content. Object identification presents one of the biggest challenges with recording and replaying Web 2.0 applications. The dynamic nature of these sites can cause the macro to fail to locate the object.

The Web Macro Recorder includes sophisticated mechanisms to overcome this challenge, including the **Highlight**, **Improve Object Identification**, **Replace**, and **Related Object** options within steps that have objects. Using these options requires that you select an object in the application. For cases where various actions are required in the application to make the object visible, such as mouse over and mouse click, use the Ctrl+Alt+F4 option to suspend the object-selection mode until you bring the object into view and press Ctrl+Alt+F4 again to select the object.

When identifying objects for applications recorded in windows, use the **Windows** tab to make sure that the correct window is selected.

After you perform any of the changes, first replay the single failed step in question, and then replay the entire macro. This will help verify whether the change has solved the issue you encountered.

The following topics describe ways to resolve object identification issues:

- "Highlighting an Object" below
- "Improving Object Identification" on the next page
- "Using Alternative Steps" on the next page
- "Modifying the Object Identification Method" on page 279
- "Modifying the Macro Timing" on page 281
- "Relating Objects to Other Objects" on page 281
- "Replacing an Object" on page 282

Highlighting an Object

For help in identifying an object previously selected in a step:

1. In the TruClient sidebar, hover your mouse over the step that contains the object.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor.
3. Click (expand) **Object**.
4. Click **Highlight** to identify the object in the application.

   If the object is found, it is temporarily highlighted by a blinking box.

   If the object is not found, an error message is displayed. For more information, see "Improving Object Identification" on the next page.

   Tip: The error could be an issue of pacing and timing, or it might indicate that the correct page to find the object is not currently displayed.

Improving Object Identification

If highlighting an object fails, you can use the improve object identification function to re-select the target object.

To re-select the object:

1. In the Step Editor for the failed step, click the **Improve object identification** icon (⬚) next to the **ID Method** field.

   The Web Macro Recorder relearns the properties of the object and compares them to the properties learned during recording. Based on the detected differences, you can make the necessary adjustments. Depending on how dynamic the application is, you may need to use the improve object identification function more than once.
2. Replay the step to see whether the problem was solved.

Using Alternative Steps

Alternative steps allow you to view multiple ways to perform the same action in a step, where it is possible. You can modify the step for the best or most consistent macro performance, or for debugging purposes.

For example, you may click on an option in a list in which the text changes based on some value. If you try to click based on the text, the step may fail. If you use an alternative step that selects the item in the list based on the ordinal value of the option within the list, the click succeeds regardless of the text.

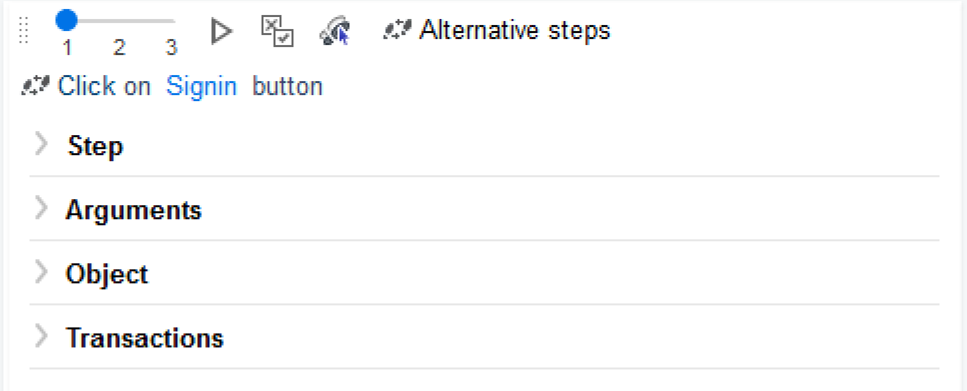
Steps that have alternative options are labeled with an alternative step icon (⁂) on the left.

Viewing and Selecting Alternative Steps

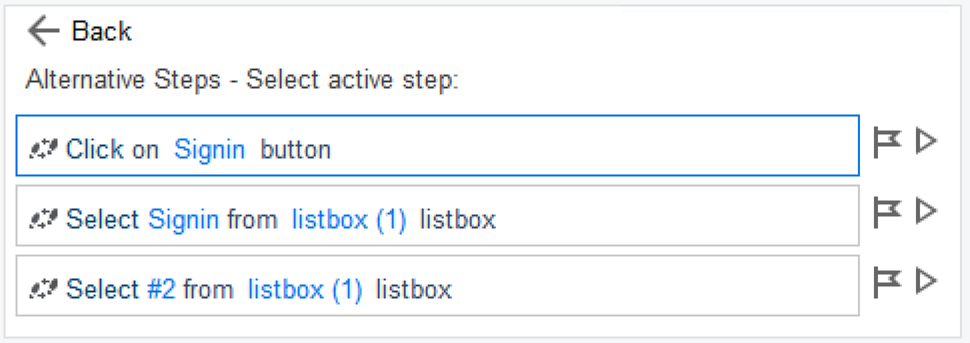
To view and select alternative steps:

1. Click the alternative step icon (⁂) to view the alternative options for that step.

   Tip: If the Step Editor is open, a button labeled **Alternative steps**, with the same icon, appears in the step's toolbar and performs the same function.

The alternative steps are shown.



← Back

Alternative Steps - Select active step:

Click on Signin button

Select Signin from listbox (1) listbox

Select #2 from listbox (1) listbox

2. Do one of the following:
   - To view an alternative step in the application, click the **Highlight the object in the AUT** icon (⚑) to the right of the alternative.

     Tip: AUT means application under test.

     This performs the same highlighting function as described in "Highlighting an Object" on page 277, with the convenience of allowing you to highlight each alternative one at a time within the macro step.

   - To play an alternative step in the application, click the **Play** icon (▷) to the right of the alternative.
3. Click an alternative to make it active.
4. Click **Back** to return to the Step Editor.

   The alternative that you selected is displayed for the step.
5. Replay the macro to test it.

Modifying the Object Identification Method

You can modify the way the Web Macro Recorder identifies the object by modifying the object identification method (ID method) in the Object section of the Step Editor.

Available Methods

The following table describes the available methods of object identification.

Method	Description
Automatic	The Automatic method is the default and recommended object identification method. This method allows the Web Macro Recorder to use its internal advanced algorithms to locate the object. **Tip:** If this method does not successfully find the object during replay, click

Method	Description
	the **Improve object identification** icon () and replay the macro again.
XPath	If Automatic identification fails, even after using improve object identification or related objects, try using the XPath identification method. This method identifies the object based on an XPath expression that defines the object in the DOM tree. For example, if you need to select the first search result, regardless of the term being searched for, using XPath identification may help. **Tip:** For the XPath ID method, the icon function changes to **Regenerate expression**. When you click the icon, you can select an object in the interface and create its associated XPath.

JavaScript	This method uses JavaScript code that returns an object. For example: `document.getElementById("SearchButton")` returns an element that has a DOM ID attribute of "SearchButton."
	This method allows you to write JavaScript code that references the returned document. You can use CSS selectors and other standard functions.
	For example, the page returned by the server contains multiple links with the same "title" attribute (search results) and we want the script to randomly click on one of the available links.
	Object identification for this case, using the JavaScript identification method, may look similar to the following:
	`var my_results = document.querySelectorAll('a [title="SearchResult"]');`
	`random(my_results);`

Descriptors	Enables you to identify an object by its properties in an editor. For more information, see TruClient Descriptors at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/descriptors.htm.

Selecting the Object Identification Method

To select a different object identification method:

1. In the TruClient sidebar, hover your mouse over the step that contains the object.
   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor for the step.
3. Click (expand) **Object**.
4. Select a different method from the **ID Method** drop-down list.
5. Continue as follows:
   - If you selected **Automatic**, the procedure is complete.
   - If you selected **XPath**, a code snippet appears in an XPath text box below the ID Method list. Optionally, click the drop-down arrow next to the **XPath** text box and select a suggested XPath code for the object.

     > **Tip:** You can click the Edit icon (✎) at the right end of the XPath text box to open the XPath Editor and edit the suggested XPath code.

   - If you selected **JavaScript**, a code snippet appears in a JavaScript text box below the ID Method list. Optionally, click the Edit icon (✎) at the right end of the **JavaScript** text box to open the JavaScript Editor and edit the suggested JavaScript code.
   - If you selected **Descriptors**, an empty Descriptors text box appears below the ID Method list. Click the Edit icon (✎) to create descriptor conditions for the object. For more information, see TruClient Descriptors at https://admhelp.microfocus.com/tc/en/12.60-12.62/Content/TruClient/descriptors.htm.

Modifying the Macro Timing

Sometimes objects may not be found because of timing and synchronization issues. For example, the macro may be looking for an object that was in the application, but the macro replayed too quickly and already progressed to another page. If you suspect that the object is not being found because of a timing or synchronization issue, you can insert Wait steps. For more information, see "Inserting a Wait Step" on page 275.

Relating Objects to Other Objects

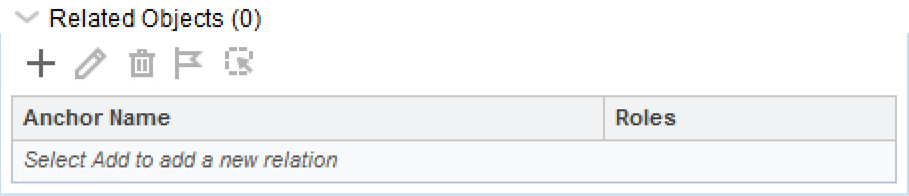
If other options do not solve the issue with object identification, you can try using the **Related Objects** option.

If an object becomes difficult to identify on its own, you can label the object based on a different, more stable object. For example, you can select an object that is not dynamic and "relate it" to the target object. Relations are defined visually, relating objects according to their distance in pixels from other objects. Relations are defined per ID method, per object. If more than one relation is defined for an ID method of a given object, both relations must locate the same object for the step to pass.

To use this function:

1. In the Step Editor for the failed step, click (expand) **Object**.
2. Click (expand) **Related Objects**.

A relation table appears.



Related Objects (0)

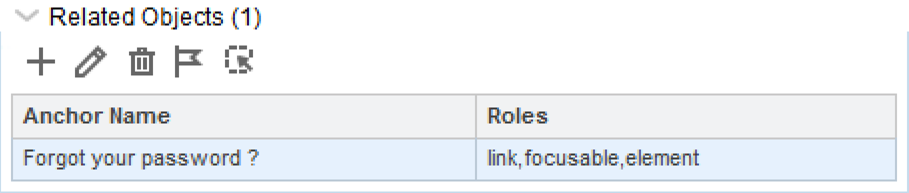
Anchor Name	Roles
Select Add to add a new relation	

3. Click +.

   The Add related object window appears.

4. Follow the onscreen directions to create a relation.

   The anchor object is added to the related object table.



Related Objects (1)

Anchor Name	Roles
Forgot your password ?	link,focusable,element

Tips

Follow these tips when using the Related Objects option:

- Use this feature only if other identification methods have failed, as it may be more resource intensive.
- Use the minimum search area to improve performance.
- Related Objects are sensitive to window sizing. Resizing may alter object positions and relationships. Take this into account.
- Each identification method (Automatic, XPath, JavaScript, and Descriptors) has its own set of related objects. These related objects are not shared among identification methods.
- If several relations exist, they must all be found in order for the identification to succeed.

Replacing an Object

If you selected the wrong object during recording, or an object has permanently changed, you can replace it with a different object without replacing the step. This effectively resets the step, deleting changes (such as relations) made to the original step.

Using the Replace option tells the macro recorder that the object currently referenced in the step is incorrect. The macro recorder removes all current knowledge of the object and learns the object you select. Therefore, you must use the Replace option only if you used the wrong object during recording.

To replace an object:

1. In the TruClient sidebar, hover your mouse over the step that contains the object.

   The Step Editor icon becomes visible.

2. Click the **Step Editor** icon (›) to open the Step Editor.
3. Click (expand) **Object**.
4. Click **Replace**.
5. Select the new object.
6. Replay the macro.

Configuring Settings

You can configure browser settings and interactive settings in the TruClient General Settings.

Accessing the TruClient General Settings

To access the General Settings:

1. In the TruClient sidebar, click the **General Settings** icon (⚙).
   The TruClient General Settings window appears.
2. Configure settings as described in "Browser Settings" below and "Interactive Options" on page 286.
3. Click Done to save the settings and close the TruClient General Settings window.

Browser Settings

The following table describes the options on the **Browser Settings** tab.

Setting	Description
Customize Keep-Alive timeout value	If you select the checkbox to enable this setting, configure the following: • **Keep-Alive timeout (milliseconds)** - Specifies the timeout (in milliseconds) for keeping idle connections open. This setting applies to both direct and proxied connections.

User Agent	Specifies the user agent string for the browser. The following list shows sample values, but is not complete:
	Default TruClient User-Agent for Windows 2016
	`Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:58.0) Gecko/20100101 Firefox/58.0`
	Windows 10-based PC Using an Edge Browser
	`Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)`

Setting	Description

	`Chrome/42.0.2311.135 Safari/537.36 Edge/12.246`
	Windows 7-based PC Using a Chrome Browser
	`Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/47.0.2526.111 Safari/537.36`

Temporary Internet Files	The browser stores copies of web pages, images, and media for faster viewing later. Configure the **check for newer versions of stored pages** to determine when the browser is to compare the local copy of resource (cache) to the Web server. The options are: • **Every time I visit the webpage** - The browser checks the resource on every request to see whether the page changed since you last viewed it. If the page has changed, the browser displays the new page and stores it in the Temporary Internet Files folder. • **Every time I start browser** - The browser checks the resource on browser start. When you view a website that you have visited before in the same browser session, the browser uses the cached temporary Internet files instead of downloading the page. • **Automatically** - The browser checks for new content only when you return to a page that you viewed in an earlier session or on an earlier date. Over time, if the browser determines that images on the page are changing infrequently, it checks for newer images less frequently. • **Never** - The browser does not check the Web server for newer content.
HTTP Version Settings	Select the **Use HTTP 1.1 through proxy connections** option to use HTTP 1.1 in connections that use proxy.
SSL	Specifies the secure connection settings. The **Minimum supported secure protocol** options are: • **SSL 3.0** - Use Secure Sockets Layer (SSL) 3.0 for minimum version of secure connections. • **TLS 1.0** - Use Transport Layer Security (TLS) 1.0 for minimum version of secure connections. • **TLS 1.1** - Use TLS 1.1 for minimum version of secure connections. • **TLS 1.2** - Use TLS 1.2 for minimum version of secure connections.

Setting	Description
Proxy	Specifies the proxy settings. The options are: • **Direct connection (proxy disabled)** - Make requests without a proxy conneciton. • **Auto detect proxy settings** - Use the Web Proxy Autodiscovery Protocol (WPAD) to locate and use a proxy autoconfig file to configure the browser's web proxy settings. • **Use System proxy settings** - Import your proxy server information from the local machine. • **Use Firefox proxy settings** - Import your proxy server information from Firefox. **Note:** Using browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used. • **Configure proxy settings using a PAC file** - Load proxy settings from a Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** field. • **Explicitly configure proxy settings** - Access the Internet through a proxy server. Provide the following server information: • **Server** - Enter the URL or IP address of your proxy server. • **Port** - Enter the port number (for example, 8080). • **Type** - Select the protocol type for handling TCP traffic through the proxy server. The options are: **Standard**, **SOCKS4**, or **SOCKS5**. • **Authentication** - If authentication is required, select a type from the Authentication list. The options are: **None**, **Basic**, **NTLM**, **Digest**, **Automatic**, **Kerberos**, or **Negotiate**. • **User Name** - If your proxy server requires authentication, enter the qualifying user name. • **Password** - If your proxy server requires authentication, enter the qualifying password. • **Bypass proxy for** - If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter

Setting	Description
	the addresses or URLs in the **Bypass Proxy For** field. Use commas to separate entries.

Interactive Options

The following table describes the settings on the **Interactive Options** tab.

Setting	Description
Enable webmacro file encryption	Encrypts the entire macro file upon saving. Otherwise, the file is saved in plain text, which exposes user names and passwords. This option is selected (ON) by default. **Note:** You can open encrypted macros even if this option is not selected. You can also open encrypted macros that were recorded using Web Macro Recorder with Firefox 30.
Action on error	Specifies the action that TruClient takes when an error occurs during replay. The options are: • **Abort script** - Abort the script on error. • **Continue to the next iteration** - Stop iteration on error and continue to the next iteration. • **Continue to the next step** - Continue to the next step on error.
Snapshot generation	Configures settings for the generation of record and replay snapshots. The **Replay snapshot generation** setting specifies whether to generate replay snapshots. The options are: • **Never** - Never generate replay snapshots. • **On error** - Generate replay snapshot when an error occurs. • **Always** - Generate replay snapshot for every step. If the **Include browser name on snapshot** option is selected, the browser name appears in the upper left corner of the snapshot.
Steps generation	Configures settings for step generation. The **Default identification method** setting options are: • **Replace server with parameter** - Replace the server name with a parameter in navigation steps.

Setting	Description
	• **Create alternative steps when applicable** - Indicate whether or not to create alternative steps (when applicable). • **Create level 2 or level 3 steps during recording** - Indicate whether or not to create steps in level 2 or level 3.
Debug	The debug settings do not apply to replay outside of debugging. The options are: • **Enable Object Identification Assistant** - Enable object identification assistant. • **Ignore wait steps** - Accelerate script debugging by ignoring wait steps. • **Hide inspector panel** - If the script hits a breakpoint, hide the inspector panel. • **Automatically populate inspector pane** - Automatically load user defined data to the inspector panel. This option does not apply to coded-action debugging.

Chapter 19: About the Web Proxy Tool

Web Proxy is a stand-alone, self-contained proxy server that you can configure and run on your desktop. With it, you can monitor traffic from a scanner, a browser, or any other tool that submits HTTP requests and receives responses from a server. It is a tool for debugging and penetration assessment; you can see every request and server response while browsing a site.

You can also create a Workflow macro or a Login macro that you can use with Fortify WebInspect.

Using Web Proxy

To use Web Proxy with a browser:

1. Click **Tools > Web Proxy**.

   The Web Proxy window opens.

   > **Note:** If you want to create a Workflow macro from a set of Burp proxy files, you can click **File > Open**, change the file type in the drop-down list from **Proxy Session File (*.Psf)** to **Burp proxy (*.*),** and then navigate to and open the Burp proxy files. See "Create a Web Macro" on page 303.

2. Click **Start** ▶ (or select **Start** from the **Proxy** menu).

   “Listening on <server:port number>” appears in the Web Proxy status bar.

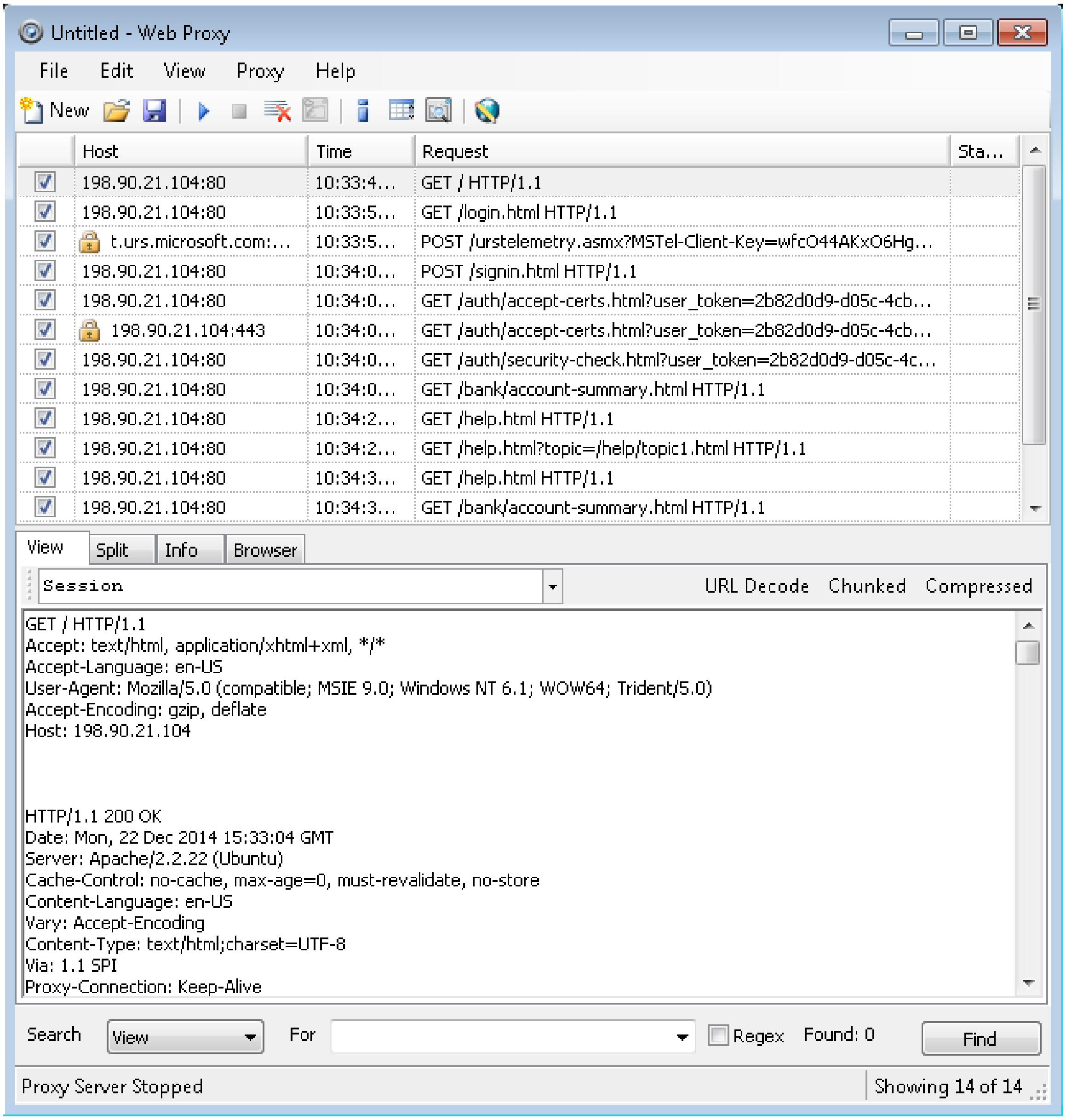
3. Click **Launch Browser** .

   This starts a Web browser and configures it to communicate through Web Proxy. Alternatively, if you prefer to use a different browser, see "Manual Configuration of Browser" on page 307 for configuration instructions.

4. Manually navigate the site for which you want capture requests/responses.
5. If Web Proxy receives a request for a certificate from a Web server, it displays a dialog box asking you to locate the certificate. The program then caches your selection on a "per server" basis. Therefore, if you subsequently want to use a different certificate for a particular server, you must clear the cache by stopping and then restarting Web Proxy.
6. When you have browsed to all necessary pages, return to Web Proxy and click 🟦 (or select **Stop** from the **Proxy** menu).

Image of Web Proxy

The following image shows the Web Proxy after it has been stopped.



	Host	Time	Request	Sta...
☑	198.90.21.104:80	10:33:4...	GET / HTTP/1.1	
☑	198.90.21.104:80	10:33:5...	GET /login.html HTTP/1.1	
☑	🔒 t.urs.microsoft.com:...	10:33:5...	POST /urstelemetry.asmx?MSTel-Client-Key=wfcO44AKxO6Hg...	
☑	198.90.21.104:80	10:34:0...	POST /signin.html HTTP/1.1	
☑	198.90.21.104:80	10:34:0...	GET /auth/accept-certs.html?user_token=2b82d0d9-d05c-4cb...	
☑	🔒 198.90.21.104:443	10:34:0...	GET /auth/accept-certs.html?user_token=2b82d0d9-d05c-4cb...	
☑	198.90.21.104:80	10:34:0...	GET /auth/security-check.html?user_token=2b82d0d9-d05c-4c...	
☑	198.90.21.104:80	10:34:0...	GET /bank/account-summary.html HTTP/1.1	
☑	198.90.21.104:80	10:34:2...	GET /help.html HTTP/1.1	
☑	198.90.21.104:80	10:34:2...	GET /help.html?topic=/help/topic1.html HTTP/1.1	
☑	198.90.21.104:80	10:34:3...	GET /help.html HTTP/1.1	
☑	198.90.21.104:80	10:34:3...	GET /bank/account-summary.html HTTP/1.1	

Session

```
GET / HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: en-US
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)
Accept-Encoding: gzip, deflate
Host: 198.90.21.104



HTTP/1.1 200 OK
Date: Mon, 22 Dec 2014 15:33:04 GMT
Server: Apache/2.2.22 (Ubuntu)
Cache-Control: no-cache, max-age=0, must-revalidate, no-store
Content-Language: en-US
Vary: Accept-Encoding
Content-Type: text/html;charset=UTF-8
Via: 1.1 SPI
Proxy-Connection: Keep-Alive
```

Search View For ☐ Regex Found: 0

Proxy Server Stopped   Showing 14 of 14

7. To change the format in which the message is displayed, select one of the tabs (**View**, **Split**, **Info**, or **Browser**).

   When using the **View** or **Split** tabs, you can enable or disable URL decoding of requests and responses by selecting the **URL Decode** button. Since most Fortify WebInspect attack traffic is URL encoded, this feature makes it easier to analyze HTTP messages. To illustrate, compare the following URL encoded and decoded versions of the same GET request:

   - `GET /notes.asp?noteid=1%20union%20%20select%200%2c1%2c2%20from%20information_schema.tables%20order%20by%204%20desc%20limit%201 HTTP/1.1`

- `GET /notes.asp?noteid=1 union select 0,1,2 from information_schema.tables order by 4 desc limit 1 HTTP/1.1`

   The **Chunked** and **Compressed** buttons are enabled if a response is either chunked-encoded or compressed. This allows you to view the original response received by Web Proxy as well as the de-chunked or decompressed response.

8. To resend a request (with or without editing), select it from the list of displayed sessions and click the HTTP Editor icon (or right-click the request and select **HTTP Editor** from the context menu).
9. To clear sessions from the list, select one or more sessions and press the Delete key (or click **Edit > Clear Selected**). To clear all sessions, click **Edit > Clear All**.

   > **Note:** When you clear a session from the Web Proxy list, you also remove it from the captured data. For example, if you have 100 sessions in the list and clear 98 of them, and then save the sessions to a file, only the two remaining sessions will be included. When clearing sessions, ignore the check boxes.

Use the File menu to save selected requests to a proxy session file (.psf) and later load them for analysis (using the **File > Open** command). You can also save a sequence of requests as a Web Macro that you can use when conducting a Fortify WebInspect scan. All **File** menu commands apply to “check-marked” requests.

Saving Sessions

To save one or more sessions for later analysis:

1. Select the sessions you want to save by placing a check mark in the left column.
2. Click the **File** menu and select **Save** or **Save As**.
3. Enter a name in the **File name** box and click **Save**.

Clearing Sessions

When you clear a session from the Web Proxy list, you also remove it from the captured data. For example, if you have 100 sessions in the list and clear 98 of them, and then save the sessions to a file, only the two remaining sessions will be included.

To clear one or more sessions:

1. Select a session. For multiple sessions, use the CTRL or SHIFT keys.

   > **Note:** Note: When clearing sessions, ignore the check boxes.

2. Do one of the following:
   - Press the **Delete** key
   - Click **Edit > Clear Selected**.

To clear all sessions, click ✗ (or click **Edit > Clear All**).

Searching a Message

You can locate information in the message displayed on the View, Split, or Info tabs using the controls at the bottom of the Web Proxy window. To search a message:

1. From the **Search** list, select a tab to search.
2. In the **For** box, enter the text (or a regular expression representing the text) you want to locate.
3. If you entered a regular expression in step 2, select the **Regex** check box.
4. Click **Find**.

Note: You can also create rules that will locate information during each session, without requiring you to manually search using the above procedure. See "Settings: Search-and-Replace" on page 298 and "Settings: Flag" on page 299.

Searching All Messages

You can search all sessions for specific information. To search all messages:

1. Click the **Toggle Search View** button 🔍 on the toolbar (or select **Search** from the **View** menu).
2. Use the **Search Area** list to specify whether you want to search the entire contents of all sessions or limit the search to a particular segment.
3. In the **Search For** box, enter a regular expression representing the text you want to locate.
4. Click **Search**.

Note: You can also create rules that will locate information during each session, without requiring you to manually search using the above procedure. See "Settings: Search-and-Replace" on page 298 and "Settings: Flag" on page 299.

Changing Options

To change Web Proxy options:

1. If Web Proxy is listening, do one of the following:
   - Click the **Proxy** menu and select **Stop**
   - Click 🟦 on the toolbar.
2. Click **Edit > Settings**, and select **Proxy Servers** tab.

See "Settings: Proxy Servers" on page 295 for more information.

Web Proxy Tabs

Each HTTP session (a single request and the associated response) is listed in the top pane of Web Proxy. When you select a session, Web Proxy displays information about the session in the lower pane. The information displayed depends on which tab you select.

You can search these tabs for specific content using the controls immediately above the status bar.

View

Use the **View** tab to select which HTTP messages you want to inspect. Options available from the drop-down list immediately below the tab are:

- Session: view the complete session (both request and response)
- Request from browser to Web Proxy: view only the request made by the browser to Web Proxy
- Request to server from Web Proxy: view only the Web Proxy request to the server
- Response from server to Web Proxy: view only the server response to Web Proxy
- Response to browser from Web Proxy: view only the Web Proxy response to the browser

Split

Click the **Split** tab to create two information areas for a single session. For example, you could show the HTTP request message created by the browser (in one area) and the HTTP response generated by the server (in the second area).

Info

Use the **Info** tab to view detailed information about the requests. Information includes the number of forms found, header information, and the properties of the page.

Browser

Click the **Browser** tab to view the response as formatted in a browser.

Web Proxy Interactive Mode

Use Interactive mode to view each browser request and each server response as the messages arrive at Web Proxy. The message will not continue toward its destination until you click **Send**. This permits you to modify the message before it is delivered.

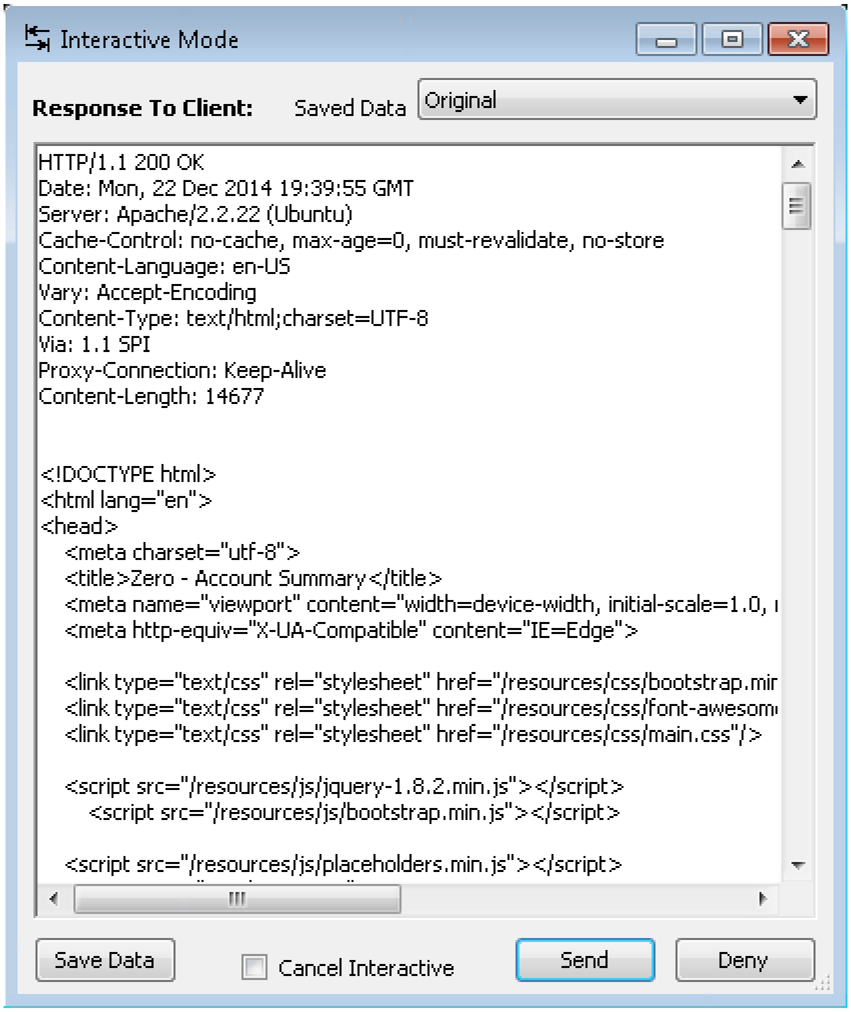
You can also prevent the message from being sent to the server by clicking **Deny**.

Using the **General** tab in the Web Proxy Settings window, you can force Web Proxy to pause as follows:

- After each request
- After each response
- After locating specific text in either the request or response (using search rules)

Image of Web Proxy Interactive Mode

The following image shows the Web Proxy in interactive mode.



```
HTTP/1.1 200 OK
Date: Mon, 22 Dec 2014 19:39:55 GMT
Server: Apache/2.2.22 (Ubuntu)
Cache-Control: no-cache, max-age=0, must-revalidate, no-store
Content-Language: en-US
Vary: Accept-Encoding
Content-Type: text/html;charset=UTF-8
Via: 1.1 SPI
Proxy-Connection: Keep-Alive
Content-Length: 14677


<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>Zero - Account Summary</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0, 
    <meta http-equiv="X-UA-Compatible" content="IE=Edge">

    <link type="text/css" rel="stylesheet" href="/resources/css/bootstrap.mir
    <link type="text/css" rel="stylesheet" href="/resources/css/font-awesom
    <link type="text/css" rel="stylesheet" href="/resources/css/main.css"/>

    <script src="/resources/js/jquery-1.8.2.min.js"></script>
        <script src="/resources/js/bootstrap.min.js"></script>

    <script src="/resources/js/placeholders.min.js"></script>
```

Enabling Interactive Mode

To enable interactive mode:

1. Click the **Proxy** menu and select **Stop**.
2. Do one of the following:
   - Click the **Proxy** menu and select **Interactive**.

- Click i on the toolbar.

3. Click the **Proxy** menu and select **Start**.

Note: When Web Proxy is in Interactive mode, a check mark appears next to the **Interactive** command on the **Proxy** menu and the Interactive icon is backlit i. Clicking the icon or selecting the command will toggle the Interactive mode on or off.

Settings

Use this property sheet to configure Web Proxy's interface, add proxy servers, and create regular expressions for locating specific information in the request or response.

Note: You cannot change settings while Web Proxy is running. Select **Stop** from the **Proxy** menu, change the settings, and then restart Web Proxy.

The Web Proxy Settings property sheet has the following tabs:

- General (see "Settings: General" below)
- Proxy Servers (see "Settings: Proxy Servers" on the next page)
- Search and Replace (see "Settings: Search-and-Replace" on page 298)
- Flag (see "Settings: Flag" on page 299)
- Evasions (see "Settings: Evasions" on page 299)

Settings: General

The **General** tab contains the following options.

Proxy Listener Configuration

Enter an IP address and port number. By default, Web Proxy uses address 127.0.0.1 and port 8080, but you can change this if necessary.

Note: Both Web Proxy and your Web browser must use the same IP address and port.

To configure Web Proxy on your host to be used by another host, you will need to change the value of the Local IP Address. The default address of 127.0.0.1 is not available to outside hosts. If you change this value to your workstation's current IP address, remote stations can use your workstation as a proxy.

Do Not Record

Use this option to create a regular expression filter that keeps files of specific types from being handled by Web Proxy. The most common types are already excluded as defaults, but other types (MPEG, PDF, etc.) can also be excluded. The purpose is to allow you to focus on HTTP request/response lines and headers by removing clutter from the message body.

Interactive

When using the interactive mode, you can force Web Proxy to pause when it:

- Receives a request from the client
- Receives a response from the server
- Finds text that satisfies the search rules you create (using the **Flag** tab)

If you select any of these options, Web Proxy will continue only after you click the **Allow** button.

Logging

Select the type of items you want to record in the log file and specify the directory in which the log file should be maintained.

If you elect to record requests and/or responses, you can also choose to convert and log the data using Base 64 encoding. This can be useful when responses contain binary data (such as images or flash files) that you want to examine.

- Raw Request refers to the HTTP message sent from the client to Web Proxy.
- Modified Request refers to the HTTP message sent from Web Proxy to the server.
- Raw Response refers to the HTTP message sent from the server to Web Proxy.
- Modified Response refers to the HTTP message sent from Web Proxy to the client.

Advanced HTTP Parsing

Most Web pages contain information that tells the browser what character set to use. This is accomplished by using the Content-Type response header (or a META tag with an HTTP-EQUIV attribute) in the HEAD section of the HTML document. For pages that do not announce their character set, you can specify which character set Web Proxy should use.

Settings: Proxy Servers

Use this area to add one or more proxy servers through which Web Proxy will route all its requests. Distributing the attack across multiple servers makes detection and counter-measures more difficult, thus mimicking how a hacker might attempt to avoid an intrusion detection system.

If you use multiple proxy servers, Web Proxy will "round-robin" the requests (i.e., Web Proxy will sequence through the list of proxy servers, sending the first request to the first server, the second request to the second server, and so on).

You can also specify IP addresses that should be accessed without using a proxy server.

Adding a Proxy Server

To add a proxy server through which Web Proxy requests will be routed:

1. In the **Proxy Address** box, type the IP address of the server through which you want to route Web Proxy requests.
2. Specify the port number in the **Proxy Port** box.
3. Select the type of proxy (standard, SOCKS4, or SOCKS5) from the **Proxy Type** list.
4. Select an authentication type: None, Auto, Kerberos, NTLM, or Basic.

   If you are unsure of which type to use, select **Auto**; Web Proxy will attempt both NTLM and Basic authentication.
5. If this server requires authentication, type your authentication credentials in the **Username** and **Password** boxes.
6. Click **Add** to add that server and display its IP address in the **Available Proxy Servers** list.

Importing a Proxy Server

To import a list of proxy servers:

1. Click **Import**.
2. Using the standard file-selection dialog box, select a delimited text file that contains the list of proxy servers.
3. Click **Open**.

The file containing proxy information must be formatted as follows:

- Each line contains one record followed by a carriage return and line feed.
- Each field in the record is separated by a semicolon.
- The fields appear in the following order:
  address;port;proxytype;username;password;authenticationtype
- The username and the password are optional. However, if authorization is not used, you must include two semicolons as placeholders.

Examples:

```
128.121.4.5;8080;Standard;magician;abracadabra;NTLM
127.153.0.3;80;socks4;;None
128.121.6.9;443;socks5;myname;mypassword;None
```

Editing Proxy Servers

To edit the list of proxy servers:

1. Select a server from the **Available Proxy Servers** list.
2. Change the information displayed in any of the controls: Proxy Address, Proxy Port, Proxy Type,

Username, or Password.

3. Click **Update**.

Removing a Proxy Server

To remove a proxy server from the list:

1. Select a server from the **Available Proxy Servers** list.
2. Click **Remove**.
3. Click **Yes** to confirm the deletion.

Bypassing Proxy Servers

If you do not need to use a proxy server to access certain URLs (such as internal testing sites), you can specify one or more hosts in the **Bypass Proxy List** area. To bypass proxy servers when accessing certain sites:

1. Click **Add**.

   The Bypass Proxy dialog box appears.
2. Enter the host portion of the HTTP URL that should be bypassed.

   Do not include the protocol (such as http://).

   For example, to bypass a proxy server for this URL

   ```
   http://zero.webappsecurity.com/Page.html
   ```
   enter this string

   ```
   zero.webappsecurity.com
   ```
   or this string

   ```
   zero.*
   ```

   > **Note:** You can also enter an IP address. Note that Web Proxy will not resolve host names to IP addresses. That is, if you specify an IP address and the HTTP request actually contains that numeric IP address, then Web Proxy will bypass a proxy server for that host. However, if the HTTP request contains a host name that normally resolves to the IP address you specify, Web Proxy will still send the request to a proxy server (unless you also specify the host name).
3. Click **OK**.

Deleting an Address

To delete an address from the **Bypass Proxy List**, select the address and click **Remove**.

Settings: Search-and-Replace

Use this tab to create rules for locating and replacing text or values in HTTP messages. This feature provides a highly flexible tool for automating your simulated attacks. Some suggested uses include:

- Masking sensitive data, such as user names and passwords
- Appending a cookie to each request
- Modifying the Accept request-header field to add or delete media types that are acceptable for the response
- Replacing a variable in the Request-URI with a cross-site scripting attack

Finding and Replacing Text

To find and replace text in requests or responses:

1. Click **Add**.

   Web Proxy creates a default entry in the table.
2. Click the **Search Field** column of the entry.
3. Click the drop-down arrow and select the message area you want to search.
4. In the **Search For** column, type the data (or a regular expression representing the data) you want to find.
5. In the **Replace With** column, type the data you want to substitute for the found data.
6. Repeat steps 1-5 to create additional search rules.

The request/response rules are applied sequentially, in the order in which they appear. For example, if a rule changes HTTPS to SSL, and if a subsequent rule then changes SSL to SECURE, the result will be that HTTPS is changed to SECURE.

Note: Search-and-replace rules are executed on request messages sent from Web Proxy to the Server and on response messages sent from Web Proxy to the Browser. You can observe the altered messages by choosing the **Info** tab, or by selecting either the **View** or **Split** tab and then choosing one of the following from the drop-down list immediately below the tab:

- Request: WebProxy -> Server
- Response: Browser <- WebProxy
- Session

Deleting a Rule

To delete a rule:

1. Select the rule you want to delete.
2. Click **Remove**.

Editing a Rule

To edit a rule:

1. Click an entry in the **Search Field**, **Search for**, or **Replace with** column.
2. Change the data

Deactivating a Rule

To deactivate a rule without deleting it:

1. Clear the **On** check box.
2. Click **OK**.

Settings: Flag

You can search areas of request and response messages to find and highlight the data you specify.

1. Click **Add**.

   Web Proxy creates a default entry in the table.
2. Click the **Search Field** column of the entry.
3. Click the drop-down arrow and select the message area you want to search.
4. In the **Search** column, type the data (or a regular expression representing the data) you want to find.
5. Click the **Flag** column of the entry.
6. Click the drop-down arrow and select a color with which to highlight the data, if found.
7. Repeat steps 1-6 to create additional search rules.

Settings: Evasions

Evasions are techniques that Web Proxy uses to circumvent intrusion detection systems, monitors, sniffers, firewalls, log parsers, or any device that attempts to shield systems from attack by filtering HTTP requests. Typically, these filters examine portions of the request, searching for "signatures" that indicate malicious threats or potential breeches of system security. If they detect these signatures, they reject the request.

To evade detection, Web Proxy modifies the HTTP request to obscure the signature for which the filter is searching, while retaining integrity sufficient for the message to be processed by the server. Of course, the techniques used by Web Proxy are not always successful. As developers become aware of methods that compromise their product's effectiveness, they incorporate procedures to combat them.

Caution! This feature is intended for use as a penetration testing tool. Do not use it or enable it when conducting vulnerability assessment scans with Fortify WebInspect.

Use the following procedure to enable evasions:

1. Select **Enable Evasions**.
2. Choose one or more evasion techniques, as described in the following sections.

Method Matching

Web Proxy replaces the GET method with HEAD. This is an attempt to defeat a filter that searches for a signature that begins with GET.

For example, the browser sends the following message to Web Proxy:

```
GET http://www.microsoft.com/secretfile.txt HTTP/1.1
```

Web Proxy sends the following message to the server:

```
HEAD http://www.microsoft.com/secretfile.txt HTTP/1.1
```

URL Encoding

Web Proxy converts characters in the URL to a "%" followed by two hexadecimal digits corresponding to the character values in the ISO-8859-1 character set.

For example, the browser sends the following message to Web Proxy:

```
GET http://zero.webappsecurity.com/cgi-bin/filename.cgi HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET %2f%63%67%69%2d%62%69%6e%2f%66%69%6c%65%6e%61%6d%65%2e%63%67%69 HTTP/1.1
Host: zero.webappsecurity.com
```

If the device is looking for "cgi-bin" as the signature, it does not match the string "%63%67%69%2d%62%69%6e" and so the request is not rejected.

Double Slashes

Web proxy converts each forward slash (/) into a double forward slash (//).

For example, the browser sends the following message to Web Proxy:

```
GET http://www.microsoft.com/en/us/secrets.aspx HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET //en//us//secrets.aspx HTTP/1.1
Host: www.microsoft.com
```

If the device is looking for "/secrets.aspx" as the signature, it does not match the string "//secrets.aspx" and so the request is not rejected.

Reverse Traversal

This technique attempts to disguise a request for a certain resource by interjecting references to relative directories, which equates to the original request.

For example, the browser sends the following message to Web Proxy:

```
GET http://www.TargetSite.com/cgi-bin/some.cgi HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /d/../cgi-bin/d/../some.cgi HTTP/1.1 [which equates to GET/cgi-
bin/some.cgi]
Host: www.TargetSite.com
```

Self-Reference Directories

Web Proxy uses the notation for parent directory (../) and current directory (./) to obfuscate the request.

For example, the browser sends the following message to Web Proxy:

```
GET http://www.TargetSite.com/cgi-bin/phf HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /./cgi-bin/./phf HTTP/1.1  [which equates to GET /cgi-bin/phf]
Host: www.TargetSite.com
```

Parameter Hiding

A request can contain parameters that are used to build dynamic page content. These parameters are typically used when search requests or selections are made and take
this form:

```
/anypage.php?attack=paramhiding&evasion=blackhat&success...
```

This technique is effective against a device that does not examine that portion of the request following the question mark (?). However, the parameter indicator can be used to potentially mask further relevant data.

For example, the browser sends the following message to Web Proxy:

```
GET /index.htm%3fparam=/../cgi -bin/test.cgi
```

Web Proxy sends the following message to the server:

```
GET /index.htm?param=/../cgi -bin/test.cgi
```

HTTP Misformatting

An HTTP request has a clearly defined structure:

```
Method<space>URI<space>HTTP/Version<CR><LF>
```

However, some Web servers will accept a request that contains a tab character instead of a space, as in the following:

```
Method<tab>URI<tab>HTTP/Version<CR><LF>
```

Any filter that incorporates the space (between the three components) as part of the signature for which it searches will fail to reject the request.

Long URLs

This technique is directed toward devices that do not examine the entire request string, but concentrate only on a subset of a programmable length (such as the first 50 characters). Web Proxy inserts a large number of random characters at the beginning of the request so that the operative portion of the request is pushed beyond the area normally examined by the filter.

For example, the browser sends the following message to Web Proxy:

```
GET http://zero.webappsecurity.com/ HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /YPVIFAHD[hundreds of characters]NIWCJBXZPXMP/../ HTTP/1.1
Host: zero.webappsecurity.com
```

DOS/Win Directory Syntax

A Windows-based filter that attempts to detect a specific signature (such as /cgi-bin/some.cgi) might be fooled if a backward slash is substituted for a forward slash (such as /cgi-bin\some.cgi). Windows-based Web servers convert a forward slash to a backward slash when interpreting directory structures, so the notation is valid. However, HTTP rules require the first character of a URI to be a forward slash.

NULL Method Processing

This technique injects a URL-encoded NULL character immediately after the METHOD (such as GET%00). It is designed for a filter that attempts to apply string operations on the request, and those string libraries use the NULL character to denote the end of a string. If this ploy is successful, detection of the NULL character prevents the device from examining the remainder of the message.

Case Sensitivity

This technique is designed to evade a filter that searches for a case-specific string.

For example, the browser sends the following message to Web Proxy:

```
GET http://zero.webappsecurity.com/cgi-bin/some.cgi HTTP/1.1
```

Web Proxy sends the following message to the server:

```
GET /CGI-BIN/SOME.CGI HTTP/1.1
Host: zero.webappsecurity.com
```

Create a Web Macro

You can use either the Web Macro Recorder or Web Proxy to create a Workflow macro or a Login macro.

A Workflow macro is used most often to focus on a particular subsection of an application. It specifies URLs that a Micro Focus scanner will use to navigate to the area. It may also include login information, but does not contain logic that will prevent the scanner from logging out of your application.

A Login macro is used for Web form authentication, allowing the scanner to log in to an application. You can also incorporate logic that will prevent the scanner from inadvertently logging out of your application.

Note: If you want to create a Workflow macro from a set of Burp proxy files, you can click **File > Open** in the menu bar of the Web Proxy tool, change the file type in the drop-down list from **Proxy Session File (*.Psf)** to **Burp proxy (*.*),** and then navigate to and open the Burp proxy files.

To create a Web macro using sessions captured by Web Proxy:

1. Select the sessions you want to include in the macro by placing a check mark in the left column.
2. Click the **File** menu and select **Create Web Macro**.
3. (Optional) On the Create Web Macro dialog box, select **Enable Check for Logout** and then enter a regular expression that identifies a unique text or phrase that occurs in the server's HTTP response when a user logs out or when a user who is not logged in requests access to a protected URL.

   Example: During a normal scan, the scanner begins crawling your site at the home page. If it encounters a link to another resource (usually through an <A HREF> HTML tag), it will navigate to that URL and continue its assessment. If it follows a link to a logout page (or if the server automatically "logs out" a client after a certain number of minutes), the scanner will not be able to visit additional resources where the client is required to be logged in. When this inadvertent log-out occurs, the scanner must be able to log in again without user intervention. This process hinges on the scanner's ability to recognize when it is no longer logged in.

   In some applications, if the user logs out (by clicking a button or some other control), the server responds with a unique message, such as "Have a nice day." If you specify this phrase as the server's logout signature, the scanner will search every response message for this phrase. Whenever it detects the phrase, the scanner will attempt to log in again by sending an HTTP request containing the user name and password.

   The scanner can also detect that it has logged out if the server sends a specific message in response to the scanner's attempt to access a password-protected URL. For example, the server may respond with a status code of "302 Object moved." If the scanner knows specifically what to look for in this response, the program will recognize that it has been logged out and can re-establish a logged-in state.

   Using the example above, if your server returns a message such as "Have a nice day" when a user logs out of your application, then enter "Have\sa\snice\sday" as the regular expression ("\s" is used in regular expressions to designate a space). A more likely example is where the server returns a

302 status code and references a new URL. In this case, "[STATUSCODE]302 AND [ALL]http://login.myco.com/config/mail?" might be a typical regex phrase. For tips on building a regular expression, see "Regular Expression Extensions" on the next page.

4. Enter a path and file name in the **Save Macro As** box, or click **Browse** to open a standard file-selection dialog box and name the file.
5. Click **OK**.

Client Certificates

If Web Proxy receives a request for a certificate from a Web Server, it displays a dialog box asking you to locate the certificate. The program then caches your selection on a "per server" basis. Therefore, if you subsequently want to use a different certificate for a particular server, you must clear the cache by stopping and then restarting Web Proxy.

Regular Expressions

Special characters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used. Another recommended resource is the Regular Expression Library.

Also see "Regular Expression Extensions" on the next page for information about special tags and operators that may be used.

Character	Description
\	Marks the next character as special. /n/ matches the character " n ". The sequence /\n/ matches a linefeed or newline character.
^	Matches the beginning of input or line. Also used with character classes as a negation character. For example, to exclude everything in the content directory except /content/en and /content/ca, use: /content/[^(en|ca)].*/.* . Also see \S \D \W.
$	Matches the end of input or line.
*	Matches the preceding character zero or more times. /zo*/ matches either " z " or "zoo."
+	Matches the preceding character one or more times. /zo+/ matches "zoo" but not "z."
?	Matches the preceding character zero or one time. /a?ve?/ matches the "ve" in "never."

Character	Description
.	Matches any single character except a newline character.
[xyz]	A character set. Matches any one of the enclosed characters. /[abc]/ matches the "a" in "plain."
\b	Matches a word boundary, such as a space. /ea*r\b/ matches the "er" in "never early."
\B	Matches a nonword boundary. /ea*r\B/ matches the "ear" in "never early."
\d	Matches a digit character. Equivalent to [0-9].
\D	Matches a nondigit character. Equivalent to [^0-9].
\f	Matches a form-feed character.
\n	Matches a linefeed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v]
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v]
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any nonword character. Equivalent to [^A-Za-z0-9_].

Regular Expression Extensions

Micro Focus engineers have developed and implemented extensions to the normal regular expression syntax. When building a regular expression, you can use the following tags and operators.

Regular Expression Tags

- [HEADERS]
- [COOKIES]
- [STATUSLINE]
- [STATUSCODE]
- [STATUSDESCRIPTION]
- [ALL]

- [BODY]
- [SETCOOKIES]
- [METHOD]
- [REQUESTLINE]
- [VERSION]
- [POSTDATA]
- [URI]

Regular Expression Operators

- AND
- OR
- NOT
- [ ]
- ( )

Examples

- To detect a response in which (a) the status line contains a status code of "200" and (b) the phrase "logged out" appears anywhere in the message body, use the following regular expression:

  [STATUSCODE]200 AND [BODY]logged\sout
- To detect a response indicating that the requested resource resides temporarily under a different URI (redirection) and having a reference to the path "/Login.asp" anywhere in the response, use the following:

  [STATUSCODE]302 AND [ALL]Login.asp
- To detect a response containing either (a) a status code of "200" and the phrase "logged out" or "session expired" anywhere in the body, or (b) a status code of "302" and a reference to the path "/Login.asp" anywhere in the response, use the following regular expression:

  ([STATUSCODE]200 AND [BODY]logged\sout OR [BODY]session\sexpired) OR ([STATUSCODE]302
  AND [ALL]Login.asp)

  Note that you must include a space (ASCII 32) before and after an "open" or "close" parenthesis; otherwise, the parenthesis will be erroneously considered as part of the regular expression.
- To detect a redirection response where "login.aspx" appears anywhere in the redirection Location header, use the following regular expression:

  [STATUSCODE]302 AND [HEADERS]Location:\slogin.aspx
- To detect a response containing a specific string (such as "Please Authenticate") in the Reason-Phrase portion of the status line, use the following regular expression:

  [STATUSDESCRIPTION]Please\sAuthenticate

Manual Configuration of Browser

If you do not start a Web browser by clicking **Launch Browser** on the Web Proxy toolbar, you can launch a browser outside the Web Proxy user interface. However, you must configure your browser's proxy settings. See your browser documentation for specific instructions.

Chapter 20: About the Web Service Test Designer

Web services are programs that communicate with other applications (rather than with users) and answer requests for information. Most Web services use Simple Object Access Protocol (SOAP) to send XML data between the Web service and the client Web application that initiated the information request. Unlike HTML, which only describes how Web pages are displayed, XML provides a framework to describe and contain structured data. The client Web application can readily understand the returned data and display that information to the end user.

A client Web application that accesses a Web service receives a Web Services Definition Language (WSDL) document so that it can understand how to communicate with the service. The WSDL document describes the programmed procedures included in the Web service, the parameters those procedures expect, and the type of return information the client Web application will receive.

Use the Web Service Test Designer to create a Web Service Test Design file (filename.wsd) containing the values that should be submitted when conducting a Web service scan.

Although the following procedure invokes the Web Service Test Designer from the Fortify WebInspect **Tools** menu, you can also open the designer through the Fortify WebInspect Scan Wizard by selecting **Start a Web Service Scan** from the Fortify WebInspect Start page and, when prompted, electing to launch the designer.

> **Note:** When the Web Service Test Designer is launched from the Fortify WebInspect Scan Wizard, if the WSDL has not yet been configured, the designer will automatically import the WSDL, assign "auto values" to each parameter, and invoke all operations. This does not occur when you launch the tool from the Fortify WebInspect Tools menu or from the Security Toolkit.

1. Select **Tools > Web Service Test Designer**.
2. On the startup dialog box, select one of the following:
    - **New Web Service Test** - Design a new Web Service test.
    - **Open Web Service Test** - Edit a design that you previously created.

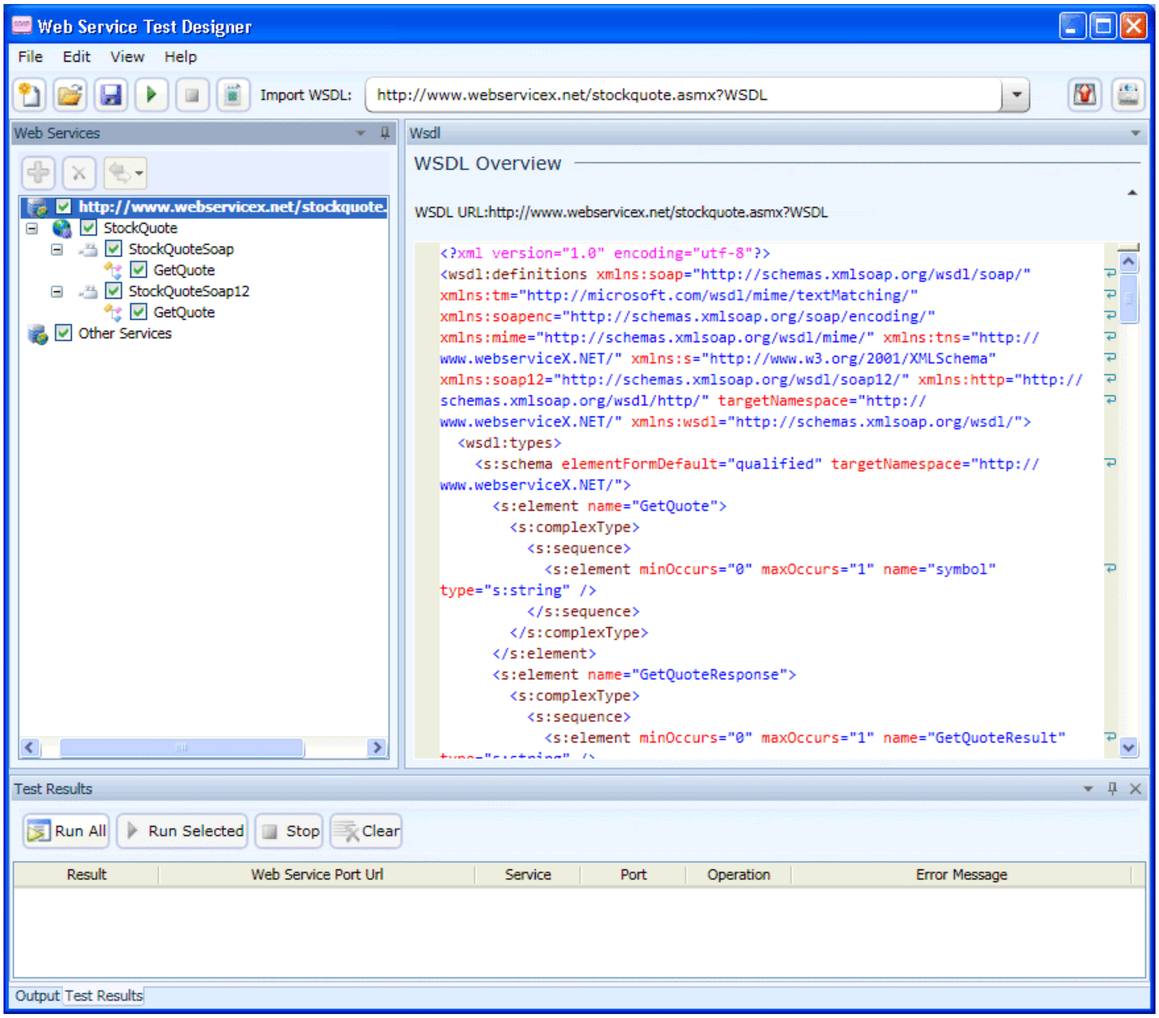
    The following procedure assumes that you are creating a design.
3. Do one of the following:
    - In the **Import WSDL** box, type or select the URL of the WSDL site (for example, http://www.webservicex.net/stockquote.asmx?WSDL) and click **Import WSDL** .
    - Click **Browse for WSDL** and select a WSDL file that you previously saved locally.

    > **Note:** If authentication is required, or if SOAP requests need to be made through a proxy server, see "Settings" on page 319 for more information.

Also note that "Other Services" appears by default. This feature is used to add services manually when a service is not associated with a WSDL. See "Manually Adding Services" on page 313 for more information. Remove the check mark next to this item.

Image of Imported WSDL

The following image shows an imported WSDL in the Web Service Test Designer.



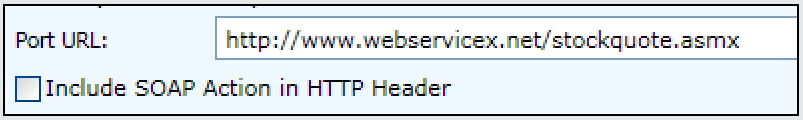
Import WSDL: http://www.webservicex.net/stockquote.asmx?WSDL

WSDL URL:http://www.webservicex.net/stockquote.asmx?WSDL

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://
www.webserviceX.NET/" xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://
schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://
www.webserviceX.NET/" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://
www.webserviceX.NET/">
      <s:element name="GetQuote">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="symbol"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="GetQuoteResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="GetQuoteResult"
type="s:string" />
```

4. Select a service transport in the left pane to display the port information in the right pane. A port defines an individual endpoint by specifying an address for a binding. Note that if the description of the WSDL includes both SOAP version 1.1 and version 1.2, and if the operations in both descriptions are the same, the versions are assumed to be identical and the services in version 1.1 only are configured. If you wish to attack both versions, then you must select the check box for each version 1.2 operation.

Note: The Port Overview panel for SOAP version 1.2 contains an additional option to include SOAP action in the HTTP header.



Port URL: http://www.webservicex.net/stockquote.asmx

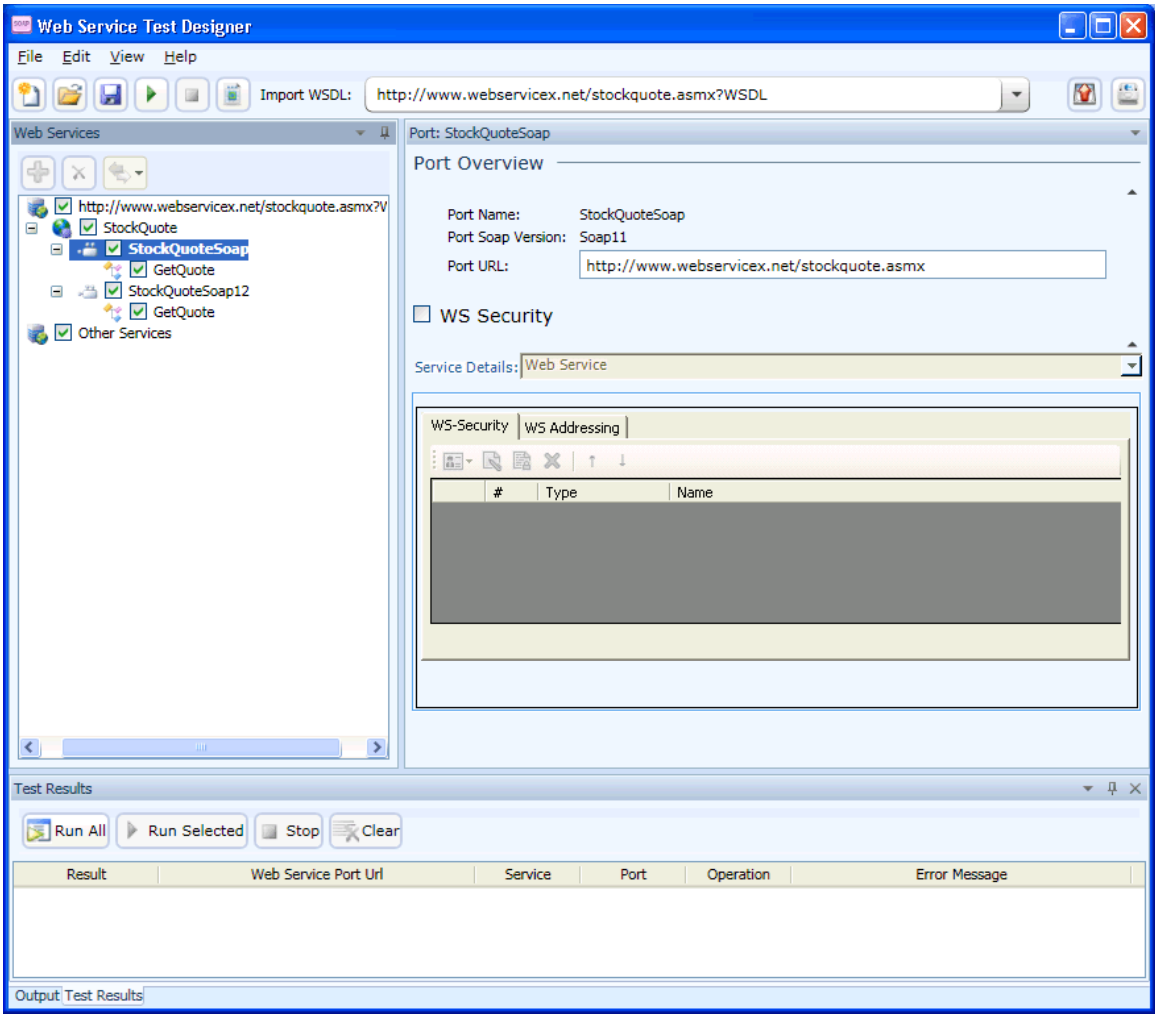
□ Include SOAP Action in HTTP Header

Even though the SOAP specification states that the SOAP Action is optional for SOAP version 1.2, some architectures require it and some cannot accept it. You can choose to include or exclude the SOAP action for a SOAP 1.2 binding, depending on your specific environment. The check box appears for SOAP 1.2 ports only and defaults to true.

Caution! RPC-encoded services require manual configuration. The **Schema Fields** tab is populated using a default SOAP schema. You can obtain the desired SOAP message from a developer or a proxy capture, and then paste the message into the **XML** tab (or import the saved message from a file). You can then click **Send** to test the operation.

Image of Service Transport/Port Information

The following image shows the port information for the selected transport.



Import WSDL: http://www.webservicex.net/stockquote.asmx?WSDL

Port Name: StockQuoteSoap
Port Soap Version: Soap11
Port URL: http://www.webservicex.net/stockquote.asmx

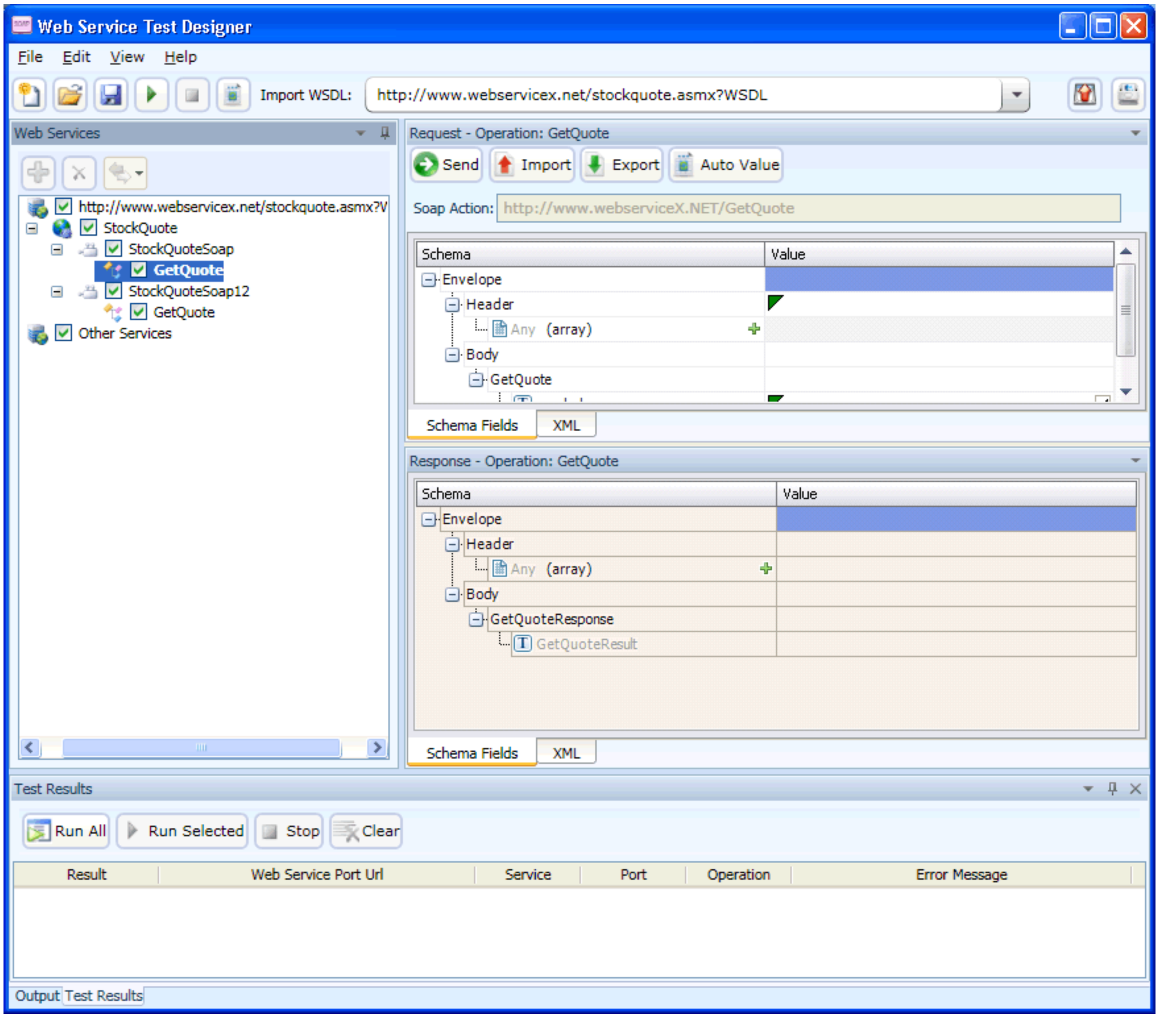
□ WS Security

Service Details: Web Service

5. If security is required:
   a. Select **WS Security**.
   b. Select an option from the **Service Details** list.
   c. Provide the required information. For help with security settings, see "WS Security" on page 323.
6. Click an operation to display schema for the request (in the top half of the right pane) and the response (in the lower half).

Image of Request/Response Schema

The following image shows the schema for the selected request.



Import WSDL: http://www.webservicex.net/stockquote.asmx?WSDL

Soap Action: http://www.webserviceX.NET/GetQuote

7. Enter a value for the operation. In this example, the user entered MFGP (the NYSE symbol for Micro Focus).

> **Note:** If you click **Auto Value**, the designer assigns a value to the operation. This value is either:
>
> - Obtained from the GlobalValuesDefault.xpr file, if the file contains an entry that matches the name of the parameter; see "Global Values Editor" on page 315 for more information.
> - Created by the designer, based on the data type. In this example, the designer would populate the parameter "symbol" with the value "symbol1."

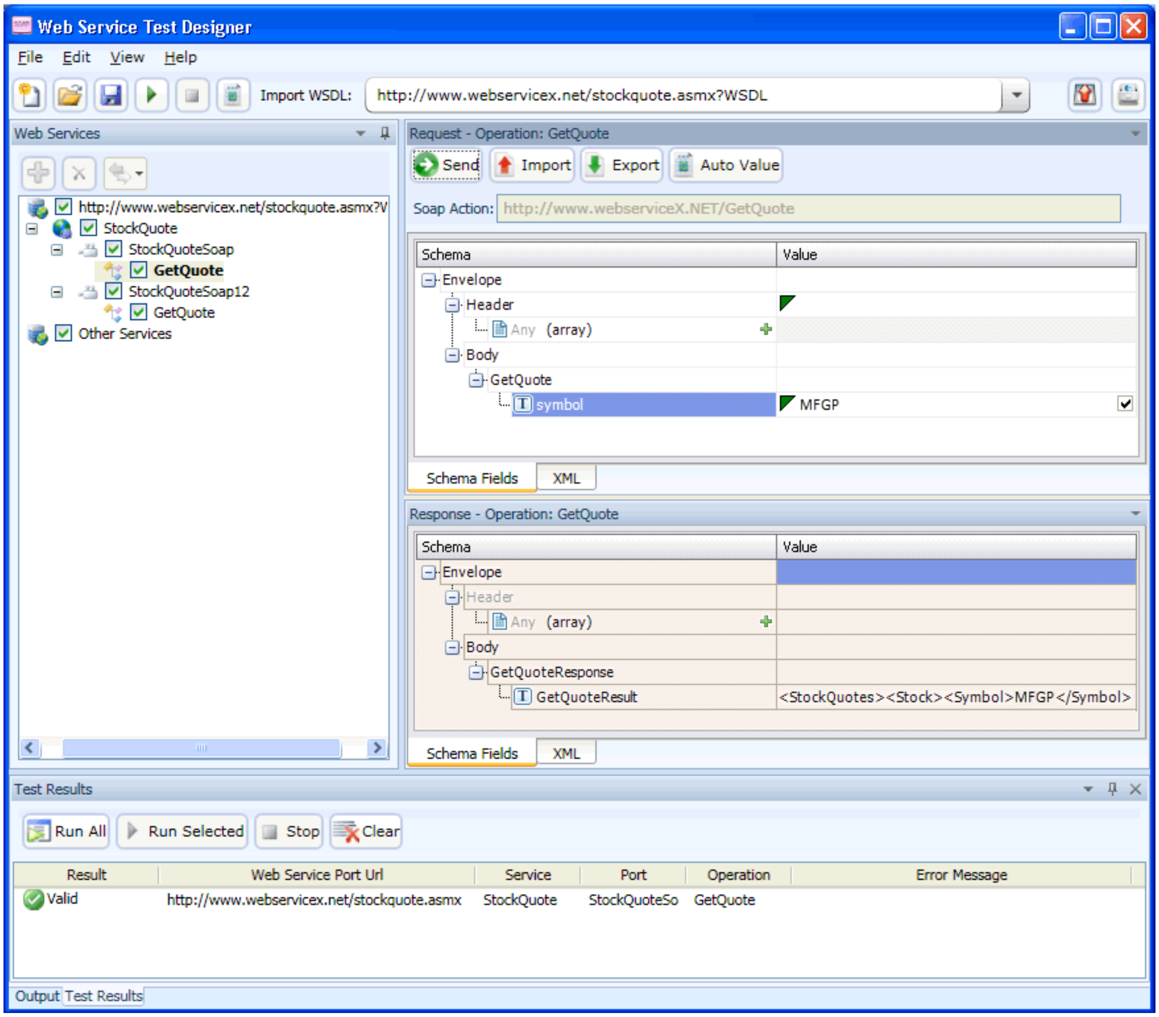
See "Using Autovalues" on page 315 for more information.

8. Click **Send**.

Results appear in the lower response pane. You can alternate between the Schema and XML views by clicking the appropriate tabs.

Image of Sending a Request

The following image shows the test results of a request that was sent.



Import WSDL: http://www.webservicex.net/stockquote.asmx?WSDL

Soap Action: http://www.webserviceX.NET/GetQuote

Schema	Value
Envelope	
Header	
Any (array)	
Body	
GetQuote	
symbol	MFGP ☑

Schema	Value
Envelope	
Header	
Any (array)	
Body	
GetQuoteResponse	
GetQuoteResult	<StockQuotes><Stock><Symbol>MFGP</Symbol>

Result	Web Service Port Url	Service	Port	Operation	Error Message
✓ Valid	http://www.webservicex.net/stockquote.asmx	StockQuote	StockQuoteSo	GetQuote	

9. When you have assigned and tested values for each operation (although only one operation is depicted in this example):
   a. Click **File > Save**.
   b. Using the standard file-selection dialog box, select a name and location for the Web Service Design file (.wsd).

> **Note:** If the WSDL contains multiple operations, data is saved for each operation regardless of whether or not the operation is checked. A check mark simply indicates that the operation will be used for auditing.

Manually Adding Services

You may encounter a Web service that does not have a WSDL associated with it.

For example, the Fortify WebInspect Recommendations module monitors scans to detect omissions, abnormalities, or anomalies that interfere with or diminish the thoroughness of a scan. If it detects SOAP requests during a Web Site scan, it suggests that you conduct a Web Service scan of that site and creates a Web Service Test Design file (*filename*.wsd) for that purpose. A WSDL file may or may not be available.

You may create a service manually, as shown in the following example.

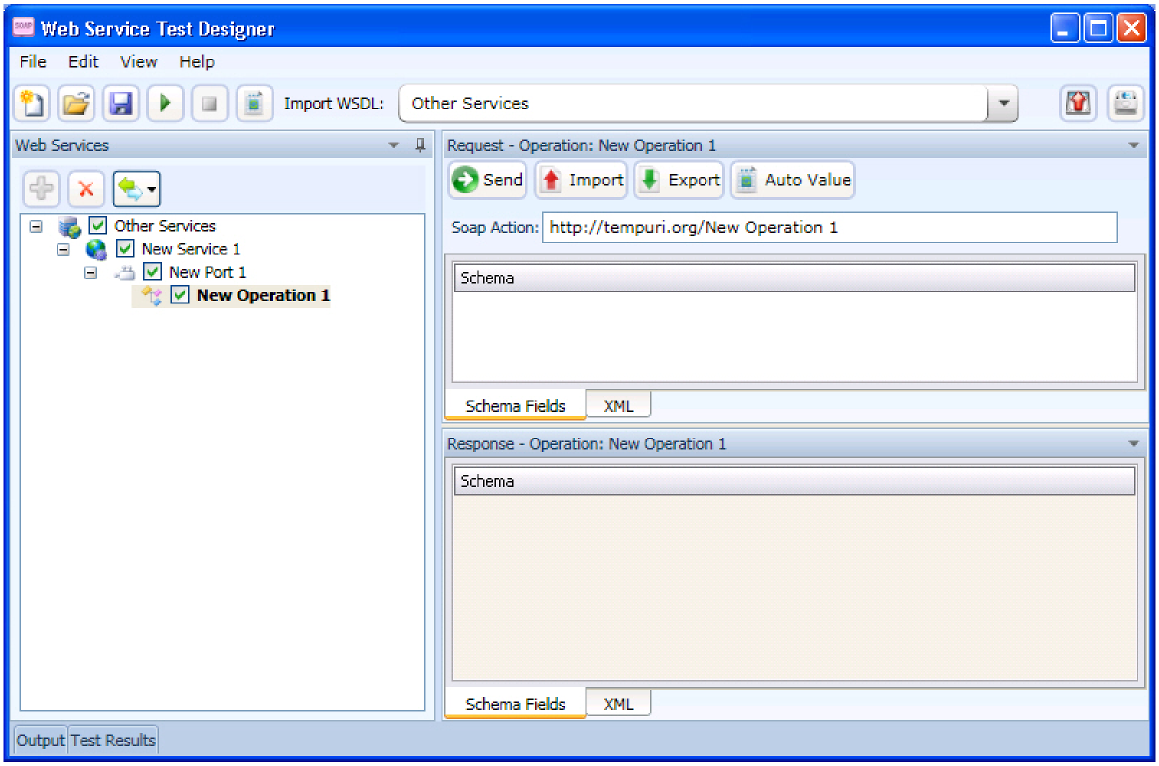
1. Right-click the default "Other Services" service and select **Add Service**.

   New Service 1 appears in the Web Services tree in the left pane.

2. If authentication is required, select **WS Security** and provide the required credentials.
3. Right-click New Service 1, select **Add Port**. and then choose either **SOAP 1.1** or **SOAP 1.2**.

   New Port 1 appears in the Web Services tree.

4. In the **Port URL box**, enter the correct URL to the service.
5. Right-click New Port 1 and select **Add Operation**.



Import WSDL: Other Services

Soap Action: http://tempuri.org/New Operation 1

> **Note:** To change service, port, or operation names, double-click the name.

6. You can import a file containing a SOAP envelope (possibly obtained using the Web Proxy tool) or you can copy and paste a SOAP envelope that you obtained from a developer onto the **XML** tab.

   If importing from a proxy capture, the SOAP action will be in the HTTP header (Soapaction=<*action_name*>).

7. If necessary, modify the values using either the **Schema Fields** tab or the **XML** tab.
8. To test the service, click either **Send** or **Run All**.

Global Values Editor

You can create a library of name/value parameters for operations that you frequently encounter. After importing a WSDL file, if you click **Set Auto Values**, the Web Service Test Designer searches the Global Values file for the names of parameters contained in the WSDL operations. If it finds a matching name, it inserts the associated value from the file into the parameter value field.

To add a global value:

1. Click **Edit > Global Values Editor**.

   The Global Values Editor opens and displays the contents of the default xml parameter registry (xpr) file named GlobalValuesDefault.xpr.
2. Click **Add**.

   This creates an entry with the default name of [Name] and a default value of [Value].
3. Click anywhere on the entry and substitute an actual name and value for the default.
4. Repeat steps 2-3 to create additional entries.
5. Do one of the following:
   - Click **OK** to save and close the file.
   - Click **Save As** to create and close the file using a different file name and/or location.

Using Autovalues

Use the Autovalues feature as an alternative to manually entering specific values for each parameter. The Web Service Test Designer analyzes each parameter and inserts a value that is likely to fulfill the service requirement. This can save considerable time when dealing with large web services.

After selecting a WSDL file:

1. Place a check mark next to each operation you want to autofill.
2. Click **Set Auto Values**.

   The following message appears: "Would you like the default values to be replaced with the defined global values?"

   If you click **Yes**, any values you may have entered manually will be erased. Also, if any parameter name in any operation matches a parameter name in the Global Values file, the associated value in the file will be substituted for the value that would normally be generated for the operation.

   If you click **No**, the function terminates.
3. Click **Yes**.
4. Click **Run All Tests** ▶.

The Web Service Test Designer submits the service request, with values inserted for each operation.

5. Click the **Test Results** tab (at the bottom of the window).
6. If an operation returned an error, double-click the operation to open it in the Request pane and manually provide a value.

See Also

"Global Values Editor" on the previous page

Importing and Exporting Operations

You can build a library of operations and their assigned values, allowing you to quickly modify other Web service designs or exchange these components with other developers/testers. Each module is saved as an XML file, such as the following request used in the preceding example:

```
<Envelope xmlns="http://www.w3.org/2003/05/soap-envelope">
  <Header />
  <Body>
    <GetQuote xmlns="http://www.webserviceX.NET/">
      <symbol>MFGP</symbol>
    </GetQuote>
  </Body>
</Envelope>
```

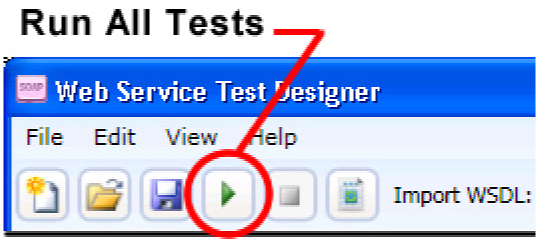
To save or import an operation:

1. Select an operation in the left pane.
2. Click **Import Request** ⬆ to load the operation.
3. Click **Export Request** ⬇ to save the operation.

Testing Your Design

You can, at any time, test the configuration of any or all operations.

After importing the WSDL, click **Run All Tests**.

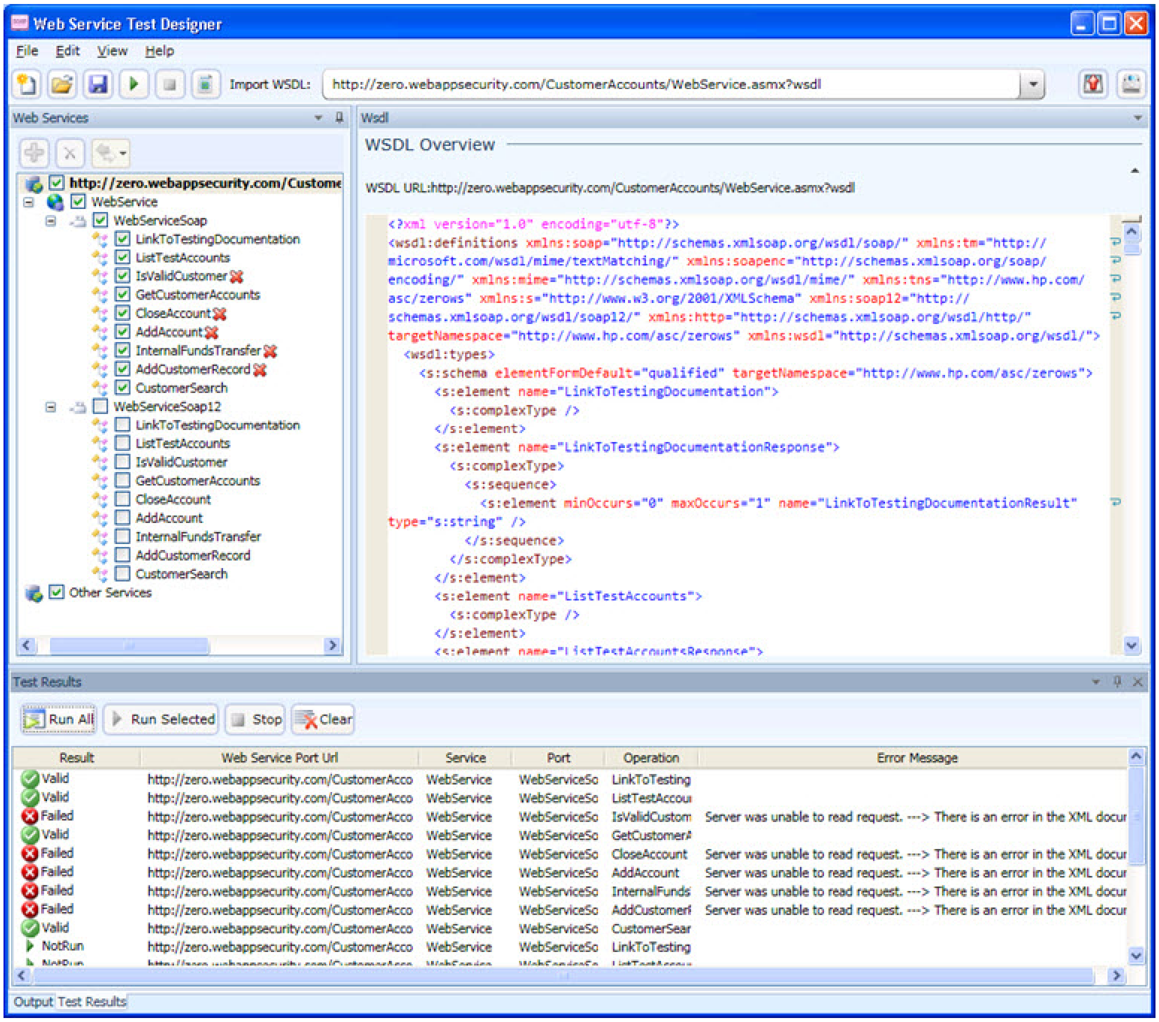


The designer attempts to submit all selected operations and displays the results.

To open the special Test Results pane, click **Test Results** on the Status bar.

Image of Test Results

The following image shows test results in the Web Service Test Designer.



WSDL URL:http://zero.webappsecurity.com/CustomerAccounts/WebService.asmx?wsdl

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tm="http://
microsoft.com/wsdl/mime/textMatching/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/
encoding/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://www.hp.com/
asc/zerows" xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://
schemas.xmlsoap.org/wsdl/soap12/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://www.hp.com/asc/zerows" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://www.hp.com/asc/zerows">
      <s:element name="LinkToTestingDocumentation">
        <s:complexType />
      </s:element>
      <s:element name="LinkToTestingDocumentationResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="LinkToTestingDocumentationResult"
type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="ListTestAccounts">
        <s:complexType />
      </s:element>
      <s:element name="ListTestAccountsResponse">
```

Result	Web Service Port Url	Service	Port	Operation	Error Message
✓ Valid	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	LinkToTesting	
✓ Valid	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	ListTestAccou	
☒ Failed	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	IsValidCustom	Server was unable to read request. ---> There is an error in the XML docur
✓ Valid	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	GetCustomerA	
☒ Failed	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	CloseAccount	Server was unable to read request. ---> There is an error in the XML docur
☒ Failed	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	AddAccount	Server was unable to read request. ---> There is an error in the XML docur
☒ Failed	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	InternalFunds	Server was unable to read request. ---> There is an error in the XML docur
☒ Failed	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	AddCustomerI	Server was unable to read request. ---> There is an error in the XML docur
✓ Valid	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	CustomerSear	
▶ NotRun	http://zero.webappsecurity.com/CustomerAcco	WebService	WebServiceSo	LinkToTesting	

The Test Results pane displays the following information:

- **Result** – The test outcome. Possible values are:
  - Valid: The operation succeeded without a server error or SOAP fault.
  - Not Run: The operation was not submitted because it was not selected (no check mark) or the Stop button was pressed before the operation was submitted.
  - Pending: The Run button has been pressed but the operation has not yet been submitted.
  - Failed: The request was unsuccessful, the server returned an error message, or a SOAP fault was received.
- **Web Service Port URL** – The URL associated with the item

- **Service** – The service associated with the item
- **Port** – The port associated with the item
- **Operation** – The operation the item represents
- **Error Message** – Explanation for failure

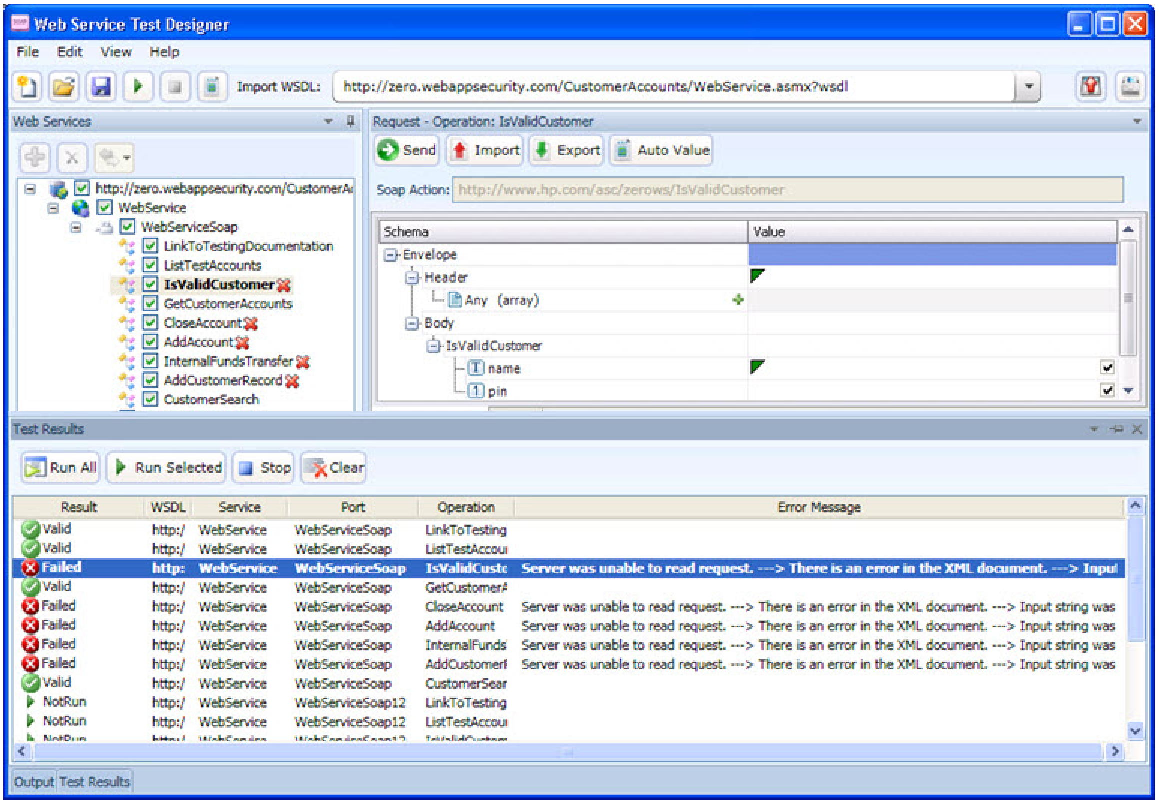
The Test Results toolbar contains the following buttons:

- **Run All** – The designer submits the service request for each checked operation.
- **Run Selected** – The designer submits the service request for operations selected in the Test Results pane.
- **Stop** – cancels the sending of service request.
- **Clear** – Removes all items from the Test Results pane.

If you double-click an item in the Test Results pane, the designer highlights the related operation in the Schema Fields pane, where you can enter values for each parameter.

Image of Selected Error with Operation Highlighted

The following image shows a selected error and its operation displayed in the Schema Fields pane.



Import WSDL: http://zero.webappsecurity.com/CustomerAccounts/WebService.asmx?wsdl

- ☑ http://zero.webappsecurity.com/CustomerA
  - ☑ WebService
    - ☑ WebServiceSoap
      - ☑ LinkToTestingDocumentation
      - ☑ ListTestAccounts
      - ☑ **IsValidCustomer** ✗
      - ☑ GetCustomerAccounts
      - ☑ CloseAccount ✗
      - ☑ AddAccount ✗
      - ☑ InternalFundsTransfer ✗
      - ☑ AddCustomerRecord ✗
      - ☑ CustomerSearch

Soap Action: http://www.hp.com/asc/zerows/IsValidCustomer

Schema	Value
Envelope	
Header	
Any (array)	
Body	
IsValidCustomer	
name	☑
pin	☑

Result	WSDL	Service	Port	Operation	Error Message
✓ Valid	http:/	WebService	WebServiceSoap	LinkToTesting	
✓ Valid	http:/	WebService	WebServiceSoap	ListTestAccou	
✗ **Failed**	**http:**	**WebService**	**WebServiceSoap**	**IsValidCusto**	**Server was unable to read request. ---> There is an error in the XML document. ---> Input**
✓ Valid	http:/	WebService	WebServiceSoap	GetCustomerA	
✗ Failed	http:/	WebService	WebServiceSoap	CloseAccount	Server was unable to read request. ---> There is an error in the XML document. ---> Input string was
✗ Failed	http:/	WebService	WebServiceSoap	AddAccount	Server was unable to read request. ---> There is an error in the XML document. ---> Input string was
✗ Failed	http:/	WebService	WebServiceSoap	InternalFunds	Server was unable to read request. ---> There is an error in the XML document. ---> Input string was
✗ Failed	http:/	WebService	WebServiceSoap	AddCustomerI	Server was unable to read request. ---> There is an error in the XML document. ---> Input string was
✓ Valid	http:/	WebService	WebServiceSoap	CustomerSear	
NotRun	http:/	WebService	WebServiceSoap12	LinkToTesting	
NotRun	http:/	WebService	WebServiceSoap12	ListTestAccou	

Settings

The Web Services Designer has two categories of settings:

- "Network Proxy" below
- "Network Authentication" on page 321

Network Proxy

To configure a network proxy:

1. Select a profile from the **Proxy Profile** list:
   - **Direct**: Do not use a proxy server.
   - **Auto Detect**: Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's Web proxy settings.
   - **Use System Proxy**: Import your proxy server information from the local machine.
   - **Use PAC File**: Load proxy settings from a Proxy Automatic Configuration (PAC) file. Then specify the file location in the **URL** box.
   - **Use Explicit Proxy Settings**: Access the Internet through a proxy server using information you provide in the Explicitly Configure Proxy section.
   - **Use Mozilla Firefox**: Import proxy server information from Firefox.

   > **Note:** Electing to use browser proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.

2. If you selected **Use PAC File**, enter the location of the PAC file in the **URL** box.
3. If you selected **Use Explicit Proxy Settings**, provide the following information:
   a. In the **Server** box, type the URL or IP address of your proxy server, followed (in the **Port** box) by the port number (for example, 8080).

   b. From the **Type** list, select a protocol for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard.

   c. If authentication is required, select a type from the **Authentication** list:

      Basic

      A widely used, industry-standard method for collecting user name and password information.

      i. The Web browser displays a window for a user to enter a previously assigned user name and password, also known as credentials.

      ii. The Web browser then attempts to establish a connection to a server using the user's credentials.

iii. If a user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials.

iv. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established.

The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers. The disadvantage is that Web browsers using Basic authentication transmit passwords in an unencrypted form. By monitoring communications on your network, an attacker can easily intercept and decode these passwords using publicly available tools. Therefore, Basic authentication is not recommended unless you are confident that the connection between the user and your Web server is secure.

NTLM

NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client’s identity without requiring that either a password or a hashed password be sent across the network.

Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and Fortify WebInspect has to pass through a proxy server to submit its requests to the Web server, Fortify WebInspect may not be able to crawl or audit that Web site. Use caution when configuring Fortify WebInspect for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

Kerberos

Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

Digest

The Windows Server 2003 operating system implements the Digest Authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

Automatic

Allow Fortify WebInspect to determine the correct authentication type.

> **Note:** Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

Negotiate

If both the server and client are using Windows 2000 or later, Kerberos authentication is used. Otherwise, NTLM authentication is used. This method is also known as Integrated Windows authentication.

4. If your proxy server requires authentication, enter the qualifying user name and password.
5. If you do not need to use a proxy server to access certain IP addresses (such as internal testing sites), enter the addresses or URLs in the **Bypass Proxy For** box. Use commas to separate entries.
6. Click **Save**.

Network Authentication

If server authentication is not required, select **None** from the **Method** list.

Otherwise, select an authentication method and enter your network credentials. The authentication methods are:

Basic

A widely used, industry-standard method for collecting user name and password information. The Web browser displays a window for a user to enter a user name and password. The Web browser then attempts to establish a connection to a server using the user's credentials. If the credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established. The advantage of Basic authentication is that it is part of the HTTP specification and is supported by most browsers.

NTLM

NTLM (NT LanMan) is an authentication process that is used by all members of the Windows NT family of products. Like its predecessor LanMan, NTLM uses a challenge/response process to prove the client's identity without requiring that either a password or a hashed password be sent across the network.

Use NTLM authentication for servers running IIS. If NTLM authentication is enabled, and Fortify WebInspect has to pass through a proxy server to submit its requests to the Web server, Fortify WebInspect may not be able to crawl or audit that Web site. Use caution when configuring Fortify WebInspect for scans of sites protected by NTLM. After scanning, you may want to disable the NTLM authentication settings to prevent any potential problem.

Automatic

Allow Fortify WebInspect to determine the correct authentication type.

Digest

The Windows Server 2003 operating system implements the Digest Authentication protocol as a security support provider (SSP), a dynamic-link library (DLL) that is supplied with the operating system. Using digest authentication, your password is never sent across the network in the clear, but is always transmitted as an MD5 digest of the user's password. In this way, the password cannot be determined by sniffing network traffic.

Kerberos

Kerberos uses the Needham-Schroeder protocol as its basis. It uses a trusted third party, termed a Key Distribution Center (KDC), which consists of two logically separate parts: an Authentication Server (AS) and a Ticket Granting Server (TGS). The client authenticates itself to AS, then demonstrates to the TGS that it is authorized to receive a ticket for a service (and receives it). The client then demonstrates to a Service Server that it has been approved to receive the service.

Using a Client Certificate

Client certificate authentication allows users to present client certificates rather than entering a user name and password. You can select a certificate from the local machine or a certificate assigned to a current user. You can also select a certificate from a mobile device, such as a common access card (CAC) reader that is connected to your computer. To use client certificates:

1. Select the **Enable client certificate on proxy** check box.
2. Click **Client Certificate**.

   The Soap Client Certificate window opens.
3. Do one of the following:
   - To use a certificate that is local to the computer and is global to all users on the computer, select **Local Machine**.
   - To use a certificate that is local to a user account on the computer, select **Current User**.

   > **Note:** Certificates used by a common access card (CAC) reader are user certificates and are stored under Current User.
4. Do one of the following:
   - To select a certificate from the "Personal" ("My") certificate store, select **My** from the drop-down list.
   - To select a trusted root certificate, select **Root** from the drop-down list.
5. Does the website use a CAC reader?
   - If *yes*, do the following:
     i. Select a certificate that is prefixed with “(SmartCard)” from the **Certificate** list.

        Information about the selected certificate and a PIN field appear in the Certificate Information area.

     ii. If a PIN is required, type the PIN for the CAC in the **PIN** field.

        > **Note:** If a PIN is required and you do not enter the PIN at this point, you must enter the PIN in the Windows Security window each time it prompts you for it during the scan.

     iii. Click **Test**.

        If you entered the correct PIN, a Success message appears.

- If *no*, select a certificate from the **Certificate** list.

  Information about the selected certificate appears in the Certificate Information area.

6. Click **OK**.

WS Security

You can configure security settings for all operations in a Web service port, using a variety of services:

- Web Service (see "Web Service Settings" below)
- Windows Communication Foundation (WCF) Service (see "WCF Service (CustomBinding) Settings" on page 325)
- WCF Service (Federation) (see "WCF Service (Federation) Settings" on page 325)
- WWCF Service (WSHttpBinding) (see "WCF Service (WSHttpBinding) Settings" on page 326)

Select an appropriate service from the **Service Details** list and then provide the requested information.

Web Service Settings

When Security credentials, known as tokens, are placed in the SOAP request, the Web server can verify that the credentials are authentic before allowing the Web Service to execute the application. To further secure Web Services, it is common to use digital signatures or encryption for the SOAP messages. Digitally signing a SOAP message verifies that the message has not been altered during transmission. Encrypting a SOAP message helps secure a Web Service by making it difficult for anyone other than the intended recipient to read the contents of the message.

WS-Security Tab

1. To add a security token, click , select a token type, and provide the requested information.
   - **UserName**. This token specifies a user name and password. You can elect to include a nonce, specify how to send the password to the server for authentication (Text, None, or Hash) and indicate whether to include a timestamp.
   - **X509 Certificate**. This token is based on an X.509 certificate. You can purchase a certificate from a certificate authority, such as VeriSign, Inc., or set up your own certificate service to issue a certificate. Most Windows servers support the public key infrastructure (PKI), which enables you to create certificates. You can then have it signed by a certificate authority or use an unsigned certificate. Select a certificate and specify the reference type (BinaryCertificateToken or Reference).
   - **Kerberos /Kerberos2**. (For Windows 2003 or XP SP1 and later). The Kerberos protocol is used to mutually authenticate users and services on an open and unsecured network. Using shared secret keys, it encrypts and signs user credentials. A third party, known as a Kerberos Key Distribution Center (KDC), authenticates the credentials. After authentication, the user may request a service ticket to access one or more services on the network. The ticket includes the encrypted, authenticated identity of the user. The tickets are obtained using the current user’s

credentials. The primary difference between the Kerberos and Kerberos2 tokens is that Kerberos2 uses the Security Support Provider Interface (SSPI), so it does not require elevated privileges to impersonate the client's identity. In addition, the Kerberos2 security token can be used to secure SOAP messages sent to a Web Service running in a Web farm. Specify the host and domain.

- **SAML Token**. Security Assertion Markup Language (SAML) is an XML standard for exchanging security-related information, called assertions, between business partners over the Internet. The assertions can include attribute statements, authentication, decision statements, and authorization decision statements. Click Load from file to browse to a SAML certificate. Click Certificate to import a certificate. Finally, select a certificate reference type: X509 Data or RSA.

2. To add a message signature, click and provide the requested information.
   - **Signing token.** The token to use for signing, usually an X.509 type. Select from the list of all added tokens.
   - **Canonicalization algorithm.** A URL for the algorithm to use for canonicalization. A drop-down list provides common algorithms. If you are unsure which value to use, keep the default.
   - **Transform algorithm.** A URL for the Transform algorithm to apply to the message signature. A drop-down list provides common algorithms. If you are unsure which value to use, keep the default.
   - **Inclusive namespaces list.** A list of comma-separated prefixes to be treated as inclusive (optional).
   - **What to sign.** The SOAP elements to sign: SOAP Body, Timestamp, and WS-Addressing.
   - **XPath (optional).** An XPath that specifies which parts in the message to sign. If left blank, the elements selected in the **Signature options** field are signed. For example, //*[local-name (.)='Body'].
   - **Token (optional).** The target token you want to sign. Select from the drop-down list of all added tokens. With most services, this field should be left empty.

3. To add message encryption, click and provide the requested information.
   - **Encrypting token.** The token to use for encryption (usually an X.509 type). You can select from a list of all previously created tokens.
   - **Encrypting type.** Indicates whether to encrypt the whole destination Element or only its Content.
   - **Key algorithm.** The algorithm to use for the encryption of the session key: RSA15 or RSAOAEP.
   - **Session algorithm.** The algorithm to use for the encryption of the SOAP message. You can select from a list of common values.
   - **XPath (optional).** An XPath that indicates the parts of the message to encrypt. If left blank, only the SOAP body is encrypted.

- **Token (optional).** The name of the encrypted token. A drop-down box provides a list of all added tokens. With most services, this field should be left empty.

4. Use the Up and Down arrows ↑ ↓ to position the security elements in order of their priority.

WS Addressing

Use the **WS-Addressing** tab to indicate whether WS-Addressing is used by the service, and if so, its version number.

WCF Service (CustomBinding) Settings

WCF Service (CustomBinding) enables the highest degree of customization. Since it is based on WCF customBinding standard, it allows you to test most WCF services, along with services on other platforms such as Java-based services that use the WS - <*spec_name*> specifications.

Transport. Select HTTP, HTTPS, or AutoSecuredHTTP. Named Pipes and TCP transport are not supported.

Encoding. Select Text, MTOM, or WCF Binary.

Security. Select an authentication mode and bootstrap policy from the appropriate list**.**

Net Security. The type of stream security: None, Windows stream security, or SSL stream security**.**

Reliable Messaging. Select **Enabled** to use reliable messaging and then select a format: either **Ordered** or **Not Ordered**.

Identities. Provide identity information for the bindings and certificate:

- **Username** and **Password**
- **Server Certificate/Client certificate**. A certificate that provides identity information for the server or client. Use the **Browse** button to open the Select Certificate dialog box.
- **Expected DNS**, **SPN**, and **UPN.** The expected identity of the server in terms of its DNS, SPN, or UPN. This can be localhost, an IP address, or a server name.

Client Windows Identity. Provide identity information for the client windows:

- **Current User**. The identity of the user logged onto the machine.
- **Custom User**. Specify the Username, Password, and Domain.

Click **Advanced** to open the Advanced Settings dialog box. See "Advanced Security Settings" on page 328 for additional information.

WCF Service (Federation) Settings

When using WCF Service (Federation), the client authenticates against the Security Token Service (STS) to obtain a token. The client uses the token to authenticate against the application server.

Server

- **Transport.** The transport type: HTTP or HTTPS.
- **Encoding.** The server's encoding policy: Text or MTOM.

Security

- **Authentication mode.** A drop-down list of possible modes of authentication, such as AnonymousForCertificate, MutualCertificate, and so forth.
- **Bootstrap Policy.** A drop-down list of possible bootstrap policies for Secure Conversation authentication, such as SspiNegotiated, UserNameOverTransport, and so forth.

Identities

The identity information for the bindings and certificate:

- **Server certificate.** A certificate that provides identity information for the server. Use the **Browse** button to open the Select Certificate dialog box.
- **Expected DNS.** The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name.

STS (Security Token Service) Details

- **Endpoint address.** The endpoint address of the STS. This can be localhost, an IP address, or a server name.
- **Binding.** The scenario which references the binding that contacts the STS.

Click **Advanced** to open the Advanced Settings dialog box. See "Advanced Security Settings" on page 328 for additional information.

WCF Service (WSHttpBinding) Settings

Using WCF Service (WSHttpBinding), you can choose from several types of authentication: None, Windows, Certificate, or Username (message protection). Select an option from the Client authentication type list. Your selection determines which additional information is required, as described below.

Type	Parameters
None	• **Negotiate server credentials**. Negotiates the Web Service's certificate with the server. You can also provide the server's DNS information. • **Specify service certificate**. The location of the service's certificate. If you select this option, the **Negotiate service credentials** option is not relevant. • **Expected server DNS.** The expected identity of the server in terms of its domain name system. This can be localhost, an IP address, or a server name. It

Type	Parameters
	can also be the common name by which the certificate was issued. • **Enable secure session**. Allows a secure session using Certificate type authentication.
Windows	• **Expected server identity**. The service principal name (SPN) or user principal name (UPN). SPN ensures that the SPN and the specific Windows account associated with the SPN identify the service. UPN ensures that the service is running under a specific Windows user account; the user account can be either the current logged-on user or the service running under a particular user account. • **Client Windows identity**. The identity information for the client windows: • **Current User.** Use the credentials of the user logged onto the machine. • **Custom User.** Provide the user credentials (Username, Password, and Domain) and optionally select an impersonation level (which determines the operations a server can perform in the client's context). Impression levels are as follows: ○ None - No level selected. ○ Anonymous - The server cannot impersonate or identify the client. ○ Identification - The server can get the identity and privileges of the client, but cannot impersonate the client. ○ Impersonation - The server can impersonate the client's security context on the local system. ○ Delegation - The server can impersonate the client's security context on remote systems. • **Enable secure session**. Allows a secure session using Windows type authentication.
Certificate	• **Client certificate**. The location of the client certificate. The **Browse** button opens the Select Certificate dialog box. • **Negotiate server credentials**. Negotiates the Web Service’s certificate with the server. You can also provide the server’s DNS information. • **Specify service certificate**. The location of the service’s certificate. If you select this option, the **Negotiate server credentials** option is disabled. • **Expected server DNS.** The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name. It can also be the common name by which the certificate was issued.

Type	Parameters
	• **Enable secure session**. Allows a secure session using Certificate type authentication.
User Name (Message Protection)	• **Username, Password**. The authentication credentials of the client. • **Negotiate server credentials**. Negotiates the Web Service's certificate with the server. You can also provide the server's DNS information. • **Specify service certificate**. The location of the service's certificate. If you select this option, the **Negotiate server credentials** option is disabled. • **Expected server DNS**. The expected identity of the server in terms of its DNS. This can be localhost, an IP address, or a server name. It can also be the common name by which the certificate was issued. • **Enable secure session**. Allows a secure session using Username type authentication.

Advanced Security Settings

This dialog box allows you to customize the security settings for your test on the following tabs.

Encoding Tab

The Encoding tab includes the following options:

- **Encoding**. The encoding type to use for the messages: Text, MTOM, or WCF Binary.
- **WS-Addressing version**. The version of WS-Addressing for the selected encoding: None, WSA 1.0, or WSA 04/08.

Advanced Standards Tab

The Advanced Standards tab includes the following options:

- **Reliable messaging**. Enables reliable messaging for services that implement the WS-ReliableMessaging specification. The encoding type to use for the messages: Text, MTOM, or WCF Binary.
- **Reliable messaging ordered**. Indicates whether the reliable session should be ordered.
- **Reliable messaging version**. The version to apply to the messages: WSReliableMessagingFebruary2005 or WSReliableMessaging11.
- **Specify via address**. Sends a message to an intermediate service that submits it to the actual server. This may also apply when you send the message to a debugging proxy. This corresponds to the WCF clientVia behavior. This is useful to separate the physical address to which the message is actually sent, from the logical address for which the message is intended.

- **Via address**. The logical address to which to send the message. It may be the physical of the final server or any name. It appears in the SOAP message as follows:

  `<wsa:Action>http://myLogicalAddress<wsa:Action>`
  The logical address is retrieved from the user interface. By default, it is the address specified in the WSDL. You can override this address using this field.

Security Tab

The Security tab includes the following options:

- **Enable secure session**. Establish a security context using the WS-SecureConversation standard.
- **Negotiate service credentials**. Allow WCF proprietary negotiations to negotiate the service's security.
- **Default algorithm suite**. The algorithm to use for symmetric/asymmetric encryption. The list of algorithms is populated from the SecurityAlgorithmSuite configuration in WCF.
- **Protection level**. Indicates whether the SOAP Body should be encrypted/signed. The possible values are: None, Sign, and Encrypt And Sign (default)
- **Message protection order**. The order for signing and encrypting. Choose from: Sign Before Encrypt, Sign Before Encrypt And Encrypt Signature, Encrypt Before Sign.
- **Message security version**. The WS-Security security version. You can also indicate whether to require derived keys for the message.
- **Security header layout**. The layout for the message header: Strict, Lax, Lax Timestamp First, or Lax Timestamp Last.
- **Key entropy mode**. The entropy mode for the security key. The possible values are: Client Entropy, Security Entropy, and Combined Entropy.
- **Require security context cancellation**. Indicates whether to require the cancellation of the security context. If you disable this option, stateful security tokens will be used in the WS-SecureConversation session, if they are enabled.
- **Include timestamp**. Includes a timestamp in the header.
- **Allow serialized signing token on reply**. Enables the reply to send a serialized signing token.
- **Require signature confirmation**. Instructs the server to send a signature confirmation in the response.

  Note: The next four options apply only when using an X.509 certificate.

- **X509 Inclusion Mode**. Specifies when to include the X.509 certificate: Always to Recipient. Never, Once, Always To Initiator.
- **X509 Reference Style**. Specify how to reference the certificate: Internal or External.
- **X509 require derived keys**. Indicates whether X.509 certificates should require derived keys.
- **X509 key identifier clause type**. The type of clause used to identify the X.509 key: Any, Thumbprint, Issuer Serial, Subject Key Identifier, Raw Data Key Identifier.

HTTP & Proxy Tab

The HTTP and Proxy tab includes the following options:

- **Transfer mode**. The transfer method for requests/responses. The possible values are Buffered, Streamed, Streamed Request, and Streamed Response.
- **Max response size (KB)**. The maximum size of the response before being concatenated.
- **Allow cookies**. Indicates whether to enable or disable cookies.
- **Keep-Alive enabled**. Indicates whether to enable or disable keep-alive connections.
- **Authentication scheme**. The HTTP authentication method: None, Digest, Negotiate, NTLM, Integrated Windows Authentication, Basic, or Anonymous.
- **Realm**. The realm of the authentication scheme in the form of a URL.
- **Require client certificate**. Indicates whether to require a certificate for SSL transport.
- **Use default web proxy**. Indicates whether to use machine’s default proxy settings.
- **Bypass proxy on local**. Indicates whether to ignore the proxy when the service is on the local machine.
- **Proxy address**. The URL of the proxy server.
- **Proxy authentication scheme**. HTTP authentication method on Proxy: Digest, Negotiate, NTLM, Basic, or Anonymous.

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Tools Guide for Fortify WebInspect Products (Fortify WebInspect Tools 19.2.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to FortifyDocTeam@microfocus.com.

We appreciate your feedback!