

HP WebInspect

for the Windows® operating system

Software Version: 9.30

User Guide

Document Release Date: September 2012
Software Release Date: September 2012



Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notice

Copyright 2012 Hewlett-Packard Development Company, L.P.

Portions Copyright ComponentOne, LLC 1991-2006.

Trademark Acknowledgements

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

Windows Vista is either a registered trademark or trademark of Microsoft Corporation in the United States and/or other countries.

Adobe and Acrobat are trademarks of Adobe Systems Incorporated.

Documentation Updates

This guide's title page contains the following identifying information:

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

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

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

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

Support

You can open a support case for Application Security Center products via e-mail or by telephone, using our new customer support system. This streamlined procedure is designed to provide easier access and improved customer satisfaction.

E-Mail (Preferred Method)

Send an e-mail to techsupport@fortify.com describing your issue. Be sure to include the product name. A customer support representative will contact you.

Telephone

Call our automated processing service at (650) 735-2215. Please provide your product name and phone number, along with a brief description of your problem. A customer support representative will contact you.

You can access the HP Application Security Community containing customer forums and blogs at :

<http://h30499.www3.hp.com/t5/Application-Security-Community/ct-p/sws-AS>

You can also visit the HP software support Web site at:

<http://support.openview.hp.com/>

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides an efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the HP Software Support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract.

To register for an HP Passport ID, go to:

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

Contents

1 Welcome to WebInspect	21
Introduction	21
Main Features of WebInspect	21
Crawling and Auditing	21
Reporting	22
Manual Hacking Control	22
Summary and Fixes	22
Scanning Policies	22
Sortable and Customizable Views	22
Enterprise-Wide Usage Capabilities	23
Web Services Scan	23
Export Wizard	23
Tools	23
Release 9.30 Features	24
New Features	24
WebInspect Enterprise Integration	24
Run a Scan in WebInspect Enterprise	24
Publish a Scan to Software Security Center	25
Allow Users to Modify the Status of Vulnerabilities Prior to Publishing	25
New 'Not Found' Tab	25
Support for IBM WebSphere Portal	26
Integration of the HP Fortify Seven Pernicious Kingdoms Taxonomy	26
Enhancements	26
Support for Security Questions with TruClient Macros	26
Automated Detection of Logout Condition with TruClient Macro Recorder	26
Easier Proxy Configuration in TruClient Macro Recorder	27
TruClient Flash Sites	27
TruClient Clear Passwords	27
False Positive Improvements	27
Secure Sockets Layer (SSL) Analyzer	27
Release 9.20 Features	28
New Features	28
Custom Parameters for URL/REST Rewriting Detection	28
Scan Comparison	28
Recommendations	28
Retest Vulnerabilities (Retest in Bulk)	28
TruClient for Login Macros	29
Software Security Center Integration	29
Common Weakness Enumeration	29

Enhancements	30
Installer/Uninstaller	30
Supportability	30
Web Services Security	30
Vulnerability Tab - Export to Excel	30
Release 9.10 Features	31
New Features	31
WebInspect Real-Time	31
Recommendations - Web Forms	32
Better Exclusion Management	32
Manual Locations	32
Enhancements	33
Flexible Vulnerability Grid	33
Simpler License Model	33
RPC-Encoded Web Services	33
Release 9.0 Features	33
New Features	33
Vulnerability Review with Retest	33
New Login Macro Recorder	34
Improved Web Services Support	35
Reuse Identified False Positives across Scans	36
Post Scan Analysis / Recommendations	36
New Blind SQL Injection Engine (BSQLI)	37
DOM-Based Cross-Site Scripting (DOM-XSS)	37
New Cross-Site Request Forgery	37
Rescan	37
Attachments and Screenshots	37
AMP Integration - Support for Assessments	38
Detachable Concurrent Licenses	38
Enhancements in Version 9.00	38
Export of Vulnerabilities Only	38
Scheduler Improvements	38
Traffic Monitor Improvements	39
Recovering Ignored Vulnerabilities and Deleted Sessions	39
Integration with the SWFScan Flash Analyzer	39
Application Settings	39
New Compliance Template	39
Easier Installation	39
Release 8.10 Features	40
Release 8.00 Features	41
New Features	41
Improvements and Enhancements	42
2 Getting Started	45
Introduction	45
Software Installation	45
Requirements	45

Installation	47
Licensing	48
Trial Registration	49
Activate Now	50
Connect to HP	51
Connect to LIM	51
License Revocation	52
Preparing Your System for Audit	52
Increased Traffic	53
Form Submission	53
E-mail Messages	53
Binary Documents	54
Sensitive Areas	54
Added Files	54
Updating WebInspect	54
Directory Structure	55
3 Using WebInspect	57
Introduction	57
Navigation Pane	59
Site View	59
Excluded Hosts	60
Allowed Hosts Criteria	61
Sequence View	61
Search View	62
Step Mode	63
Navigation Pane Icons	63
Navigation Pane Shortcut Menu	64
Information Pane	67
Scan Info Panel	67
Dashboard	68
Traffic Monitor	70
Recommendations	71
Attachments	74
False Positives	74
Session Info Panel	76
Host Info Panel	79
Summary Pane	81
Vulnerabilities Tab	82
Recovering Deleted Items	85
Retesting/Reviewing Vulnerabilities	85
Not Found Tab	86
Information Tab	86
Best Practices Tab	87
Scan Log Tab	87
Server Information Tab	87
Using Filters and Groups in the Summary Pane	88

Using Filters	88
Using Groups	90
WebInspect Toolbars	90
WebInspect Menu Bar	93
File Menu	94
Edit Menu	94
View Menu	95
Tools Menu	95
Scan Menu	95
Enterprise Server Menu	95
Reports Menu	96
Help Menu	96
HP Application Security Center	97
Inspecting Results: Web Site Scan	97
Session	99
Session Shortcut Menu	99
Editing Vulnerabilities	99
Inspecting Results: Web Service Scan	102
Exporting Scans	103
Exporting Scan Details	104
Publishing to Software Security Center	105
Publishing through WebInspect Enterprise	105
Integrating with WebInspect Enterprise	106
Publishing through AMP	108
Importing Scans	109
Running a Scan in AMP or WebInspect Enterprise	109
Uploading a Scan to an Enterprise Server	111
Managing Settings	112
Managing Scans	113
Managing Scheduled Scans	114
Scheduling a Scan	114
Selecting a Report	116
Configure Report Settings	117
Stopping a Scheduled Scan	117
Generating Reports	118
Advanced Report Options	120
Compliance Templates	121
21CFR11	121
Basel II	121
CA OPPA	122
CASB 1386	122
COPPA	122
DCID	122
DoD Application Security Checklist Version 2	123
DoD Application Security and Development STIG V3 R2	123
EU Data Protection	123

EU Directive on Privacy and Electronic Communications	123
FISMA	124
GLBA	124
HIPAA	124
ISO17799	124
ISO27001	124
JPIPA	124
NERC	125
NIST 800-53	125
OMB	125
OWASP Top Ten 2004/2007/2010	125
PCI Data Security 1.2, 2.0	125
PIPEDA	125
Safe Harbor	126
SANS CWE Top 25	126
Sarbanes-Oxley	126
UK Data Protection	126
WASC	127
License Management.....	127
Connected to HP	128
Updating the License.....	128
Deactivating the License.....	128
Configuring the License	128
Complete Offline Activation	129
Connected to License and Infrastructure Manager	130
Selecting Detached or Connected License	130
Configuring the License	130
Command Line Execution	131
Hyphens in Command Line Arguments	135
Uninstalling WebInspect	135
4 Scanning a Site	137
Introduction	137
Web Site Scan.....	137
WebSphere Portal Frequently Asked Questions.....	146
Web Service Scan	148
Enterprise Scan	151
Manual Scan.....	155
Reviewing and Retesting Vulnerabilities	156
Retest Individual Vulnerability	156
Retest All Vulnerabilities	156
Rescan the Site	157
Compare Scans	157
Dashboard	158
Vulnerability Compare	159
Severity	160
Session Info	160

Summary Pane.....	160
FilesToURLs Utility.....	162
Usage for FilesToURLs.exe	163
Usage for FilesToURLs.py	163
Internet Protocol Version 6	164
5 Default Scan Settings.....	165
Introduction	165
Scan Settings	165
Method.....	165
Scan Mode	165
Crawl and Audit Mode	166
Navigation	166
General	167
Scan Details	167
Crawl Details	169
Content Analyzers	172
Flash.....	172
JavaScript/VBScript	172
Silverlight.....	173
Recommendations.....	173
Requestor.....	174
Requestor Performance.....	175
Requestor Settings.....	175
Stop Scan if Loss of Connectivity Detected	176
Session Storage.....	176
Session Exclusions	178
Allowed Hosts	181
HTTP Parsing	182
Custom Parameters	183
URL Rewriting.....	183
RESTful Services.....	183
Enable automatic seeding of rules that were not used during scan.....	184
Double Encode URL Parameters	185
Creating Rules for Matrix and Path Parameters.....	185
Filters	188
Cookies/Headers	189
Standard Header Parameters.....	189
Append Custom Headers.....	189
Append Custom Cookies	190
Proxy	190
Authentication	192
File Not Found	194
Policy	194
Create a Policy.....	195
Import a Policy.....	195
Delete a Policy	195

Edit a Policy	196
Crawl Settings	196
Link Parsing	196
Session Exclusions	196
Audit Settings	199
Session Exclusions	199
Attack Exclusions	202
Attack Expressions	204
Vulnerability Filtering	204
Smart Scan	205
Importing and Exporting Settings	206
Managing Settings	206
Creating a Settings File	206
Editing or Deleting a Settings File	206
Exporting a Settings File	207
Transferring Settings to or from an Enterprise Server	207
6 Application Settings.....	209
Introduction	209
General	209
General Settings	209
HP SecurityScope Settings	210
Macro Recorder	210
Database	211
Directories	212
License	212
License Details	212
Direct Connection to HP	212
Connection to LIM	213
Server Profiler	213
Step Mode	215
Logging	215
Proxy	215
Reports	216
Options	216
Headers and Footers	217
Run as a Service	218
Interactive	218
Sensor	218
Smart Update	219
Support Channel	219
HP Quality Center	219
IBM Rational ClearQuest	220
7 WebInspect Tools	223
Introduction	223
Client Certificates	224

Policy Manager.....	224
Views	224
Standard View	224
Search View	225
Creating or Editing a Policy.....	226
Creating a Custom Check.....	227
Disabling a Custom Check.....	233
Deleting a Custom Check.....	233
Editing a Custom Check.....	233
Searching for Attack Agents	234
Policy Manager Icons	235
Audit Inputs Editor	236
Engine Inputs	236
Check Inputs.....	237
4719: IIS Mapping	237
4721: Admin Section Must Require Authentication	238
4722: Logins Sent Over Unencrypted Connection.....	238
4723: Logins Sent Over Query	238
4724: Password Field Masked.....	238
4726: Secure Section Only Accessible Via SSL	238
4728: Persistent Cookies	238
4729: User supplied data without POST	239
4731: Script Directory Check	239
4732: Script File Extension Disclosure.....	239
5151: Arbitrary Remote File Include	239
5546: Privacy Policy Not Present	241
10167: Password in Query or Cookie Data.....	241
10183: Allowed Top-Level Domain	241
10274: Proxy CONNECT Access.....	241
10275: Proxy GET Access	242
10280: Price-Related Form Fields.....	242
10287: Local File Include	242
10551: Possible Username or Password Disclosure.....	244
10963: Cross-Site Request Forgery.....	244
10965: User Data in Query or Cookie.....	245
Web Form Editor	246
Manually Creating a Web Form List.....	246
Recording Web Form Values.....	248
Importing a Web Form File	250
Scanning with a Web Form File.....	250
Web Form Editor Settings	251
General.....	251
Proxy.....	251
Web Form Logic.....	253
Web Brute.....	255
Mounting a Brute Force Attack	255
Creating and Importing Lists	257

Exporting Dictionaries	257
Web Brute Settings.....	258
Options.....	258
Authentication.....	259
Proxy.....	259
Web Discovery	261
Discovering Sites.....	261
Web Discovery Settings	262
Select Protocols	262
Logging.....	262
Connectivity.....	262
Encoders/Decoders	264
Encoding a String.....	264
Decoding a String.....	264
Manipulating Encoded Strings	265
Encoding Types.....	265
Prefixed	266
Regular Expression Editor	267
Testing a Regular Expression	267
Regular Expressions.....	268
Regular Expression Extensions	269
Examples	270
HTTP Editor.....	271
Request Viewer	271
Response Viewer	271
HTTP Editor Menus.....	272
File Menu	272
Edit Menu	272
View Menu	272
Help Menu	272
Request Actions	273
PUT File Upload	273
Change Content-Length	273
URL Encode/Decode Param Values	273
Unicode Encode/Decode Request	274
Create MultiPart Post	274
Remove MultiPart Post	274
Response Actions	274
Chunked.....	274
Content Codings	275
Editing and Sending Requests.....	275
Searching for Text.....	275
HTTP Editor Settings.....	275
Options.....	276
Authentication.....	278
Proxy.....	278
Web Proxy	279

Using Web Proxy.....	279
Creating a Web Macro	281
Web Proxy Tabs.....	282
Web Proxy Settings.....	283
Web Proxy Interactive Mode	289
Smart Update.....	290
Checking for Updates Automatically.....	290
Cookie Cruncher.....	291
Background	291
Using the Cookie Cruncher	291
Subcookies.....	292
Cookie Cruncher Tabs.....	293
Cookies Tab	293
Character Sets Tab	293
Char Freq Tab	294
Randomness Tab	294
Predictability Tab	294
Disk Plot Tab	295
Cookie Cruncher Settings.....	296
General.....	296
Authentication.....	297
Proxy.....	297
Web Fuzzer.....	299
Using the Web Fuzzer.....	299
Filters	300
Creating a Filter	301
Using a Filter	301
Deleting a Filter.....	301
Editing a Filter	301
Using the Session Editor	302
Creating a Query String.....	302
Session Editor Tabs	303
Method Tab	303
Path Tab.....	303
Query Tab.....	303
Version Tab.....	303
Headers Tab	303
Cookies Tab	304
Post Data Tab.....	304
Web Fuzzer Settings.....	305
General.....	305
Proxy.....	305
SQL Injector.....	307
Using the SQL Injector.....	307
SQL Injector Tabs.....	309
SQL Injector Settings.....	309
Options Tab	309

Authentication Tab	311
Proxy Tab	311
Compliance Manager	313
How It Works	313
Creating a Compliance Template.....	313
Usage Notes	316
Testing for Compliance.....	316
Log Viewer	318
Viewing Logs.....	318
Web Macro Recorder (Traffic-Mode)	319
Creating a Macro	319
Editing the Logout Condition	321
URL Rewriting and Request Parameters.....	323
Inspecting and Editing a Macro.....	324
Traffic-Mode Web Macro Recorder Settings	326
General.....	326
Proxy.....	327
Web Macro Recorder Menus.....	328
File	328
Edit	328
View	329
Help	329
Web Macro Recorder (Event-Based IE Compatible)	330
Recording a Log-In Macro	330
Specifying a Logout Condition	331
Specifying a Confirmation Element	331
Troubleshooting a Macro.....	331
Editing a macro	332
Example: Adding Elements for I-Frame Login	333
Dynamic Challenge-Response Authentication	334
Logout Elements	336
Using a Regular Expression for Logout Detection	337
Confirmation Elements (Hints)	338
Unsupported Elements.....	338
Event-Based Web Macro Recorder Settings	339
Application Settings	339
Macro Settings	340
Web Macro Recorder (TruClient)	342
Recording a Macro	342
Parameterizing Input	344
Using Name and Password Parameters	344
Using URL Parameters	345
Recording a Multi-Challenge Macro	346
Enhancing Macros	349
Debugging Macros	350
Resolving Object Identification Issues.....	351
Inserting and Modifying Loops	354

Script Levels	354
Alternative Steps	355
Snapshots	355
Toolbox.	356
Settings	356
Server Analyzer	360
Analyzing a Server	360
Server Analyzer Settings	360
Authentication Method	360
Authentication Credentials.	361
Proxy	361
Exporting Results	362
Server Profiler	363
Launching the Server Profiler as a Tool	363
Invoking the Server Profiler when Starting a Scan	364
SWFScan	365
Vulnerability Detection.	365
ActionScript 3 Vulnerabilities Detected by SWFScan.	365
ActionScript 1 and 2 Vulnerabilities Detected by SWFScan.	367
Analyzing Flash Files.	369
Analyze a Flash file using SWFScan as a standalone tool	369
Analyze a Flash file using SWFScan as an integrated component of WebInspect	370
Examining Results	370
Searching Source Code	371
SWFScan Settings	372
Report Designer	374
User Interface	374
Toolbar	374
Menus.	375
Designer Tabs	377
Toolbox	377
Design Surface.	378
Report Explorer	378
Properties Grid	379
Creating a Report	380
Report Script Editor	380
Parameter Designer	381
Toolbar	382
Canvas	382
Properties Grid Pane.	383
Controls Toolbox	383
Report Parameters Pane.	383
Report Styles Editor	383
Report Structure	384
Report Structure	384
Report Header	384
Report Footer	384

Page Header	384
Page Footer	385
Group Header/Footer	385
Detail	385
Report Settings	385
Charts	385
Chart Types	385
Chart Data	397
Chart Effects	401
Chart Control Items	404
Chart Axes and Walls	406
Chart-Specific Properties	409
Chart Wizard	410
Walk-Through: Creating a Report	410
Populate the Detail section	414
HP Support Tool	416
Scrubbing Data	417
Support Settings	418
Proxy	418
SQL Server	418
Advanced	419
Web Service Test Designer	420
WS Security Settings	423
Web Service	424
WCF Service (CustomBinding)	425
WCF Service (Federation)	426
WCF Service (WSHttpBinding)	427
Advanced Security Settings	428
Manually Adding Services	430
Global Values Editor	431
Importing and Exporting Operations	431
Using Autovalues	432
Testing Your Design	432
Web Service Test Designer Settings	435
Network Proxy	435
Network Authentication	435
Web Application Firewall Integration Tool	436
A Attacks and Methodologies	439
Introduction	439
Parameter Manipulation	440
Query strings	440
Postdata	440
Headers	440
Cookies	440
Parameter Injection	441
Command Execution	441

SQL injection	441
Cross-site scripting	442
Abnormal input	442
Hidden content access	442
Untrusted application access	443
Format string attack	443
Parameter Overflow	443
Numeric overflow	443
String overflow	444
Parameter Addition	444
Application debug/backdoor modes	444
Internal parameter specification	444
Parameter Deletion	445
Path Manipulation	445
Path truncation	445
Case sensitivity	445
Character encoding	445
MS-DOS 8.3 Short Filename	446
Directory traversal	446
Character stripping	446
Webserver Assessment	447
HTTP compliance	447
WebDAV compliance	447
SSL strength	447
Certificate analysis	447
HTTP Method Support	447
Site Searching	448
Test files	448
Administrative interfaces	448
Program dumps	448
Application logs	448
Installation documentation	448
Backup files	448
Site statistics pages	449
Application Mapping	449
Crawl	449
Automatic form filling	449
SSL support	449
Proxy support	449
Client certificate support	449
State management	449
Directory enumeration	450
Brute Force Authentication Attacks	450
Content Investigation	450
Known Attacks	451
B Policies and Components	453

Introduction	453
Policies	453
Best Practices	453
Hazardous	453
By Type	454
Custom.....	455
Policy Components	455
Audit Engines	456
General Application Testing.....	458
General Text Searching	458
Third-Party Web Applications	458
Web Frameworks/Languages.....	458
Web Servers.....	458
Web Site Discovery	459
Custom Checks	459
C HTTP Status Codes	461
Introduction	461
Glossary	465
Index	473

1 Welcome to WebInspect

Introduction

WebInspect is the most accurate and comprehensive automated Web application and Web services vulnerability scanning solution available today. With WebInspect, security professionals and compliance auditors can quickly and easily analyze the numerous Web applications and Web services in their environment. WebInspect is the only product that is maintained and updated daily by the world's leading Web security experts. These solutions are specifically designed to assess potential security flaws and to provide all the information you need to fix them.

WebInspect delivers the latest evolution in scanning technology, a Web application security product that adapts to any enterprise environment. As you initiate a scan, WebInspect assigns "assessment agents" that dynamically catalog all areas of a Web application. As these agents complete the assessment, findings are reported to a main security engine that analyzes the results. WebInspect then launches audit engines to evaluate the gathered information and apply attack algorithms to locate vulnerabilities and determine their severity. With this smart approach, WebInspect continuously applies appropriate scan resources that adapt to your specific application environment.

Main Features of WebInspect

The following is a brief overview of what you can do with WebInspect, and how it can benefit your organization.

Crawling and Auditing

WebInspect uses two basic modes for determining your security weaknesses.

- A crawl is the process by which WebInspect identifies the structure of the target Web site. In essence, a crawl runs until no more links on the URL can be followed.
- An audit is the actual vulnerability assessment. A crawl and an audit, when combined into one function, is termed a scan.

Simultaneous crawl and audit combines application crawl and audit phases into a single fluid process. The scan is refined based on real-time audit findings, resulting in a comprehensive view of an entire Web application's attack surface. Intelligent engines employ a structured, logic-based approach to analyzing an application and then customize attacks based on the application's behavior and environment. WebInspect combines these sophisticated, ground-breaking scanning technologies with known Web application vulnerabilities that are stored in a vulnerability database.

Reporting

Use WebInspect reports to gain valuable, organized application information. You can customize report details, deciding what level of information to contain in each report, and gear the report for a specific audience. You can also create your own reports, using the Report Designer. You can save reports in a variety of formats, and you can also include graphic summaries of vulnerability data. For more information, see [Generating Reports](#) on page 118.

Manual Hacking Control

With WebInspect, you can see what's really happening on your site, and simulate a true attack environment. WebInspect functionality gives you the ability to view the code for any page that contains vulnerabilities, then make changes to server requests and resubmit them instantly. For more information, see [HTTP Editor](#) on page 271.

When using the Web Proxy tool, you can also pause the client-server data flow when Web Proxy receives a request from the client, receives a response from the server, or finds text that satisfies the search rules you create. For more information, see [Web Proxy Settings](#) on page 283.

Summary and Fixes

WebInspect provides summary and remediation information for all vulnerabilities detected during a scan. This includes reference material, links to patches, instructions for prevention of future problems, and vulnerability solutions. As new attacks and exploits are formulated, we update our summary and fix information database. Use Smart Update on the WebInspect toolbar to update your database with the latest vulnerability solution information.

Scanning Policies

You can edit and customize scanning policies to suit the needs of your organization, reducing the amount of time it takes for WebInspect to complete a full scan. For more information on configuring WebInspect policies, see [Policy Manager](#) on page 224.

WebInspect also lets you extend the product's capabilities to meet your organization's specific needs. You can configure HP WebInspect to adapt to any web application environment and use the custom check wizard to create custom attacks.

Sortable and Customizable Views

When conducting or viewing a scan, the navigation pane on the left side of the WebInspect window includes the Site, Sequence, Search, and Step Mode buttons, which determine the contents (or "view") presented in the navigation pane.

- Sequence view displays server resources in the order they were encountered by WebInspect during an automated scan or a manual crawl (Step Mode).
- Search view allows you to locate sessions that fulfill the criteria you specify.
- Site view presents the hierarchical file structure of the scanned site.
- Step Mode is used to navigate manually through the site, beginning with a session you select from either the site view or the sequence view.

Enterprise-Wide Usage Capabilities

The integrated scan process provides a comprehensive overview of your Web presence from an overall enterprise perspective, enabling you to selectively conduct application scans, either individually or scheduled, of all Web-enabled applications on the network.

Web Services Scan

WebInspect can provide a comprehensive scan of your Web services vulnerabilities, allowing you to assess applications containing Web services.

Export Wizard

WebInspect's configurable XML export tool enables users to export (in a standardized XML format) any and all information found during the scan. This includes comments, hidden fields, JavaScript, cookies, Web forms, URLs, requests, and sessions. Users can specify the type of information to be exported. The Export Wizard also includes a "scrubbing" feature that prevents any sensitive data from being included in the export.

Tools

A robust set of diagnostic and penetration testing tools is packaged with WebInspect. These include:

- Audit Inputs Editor
- Compliance Manager
- Cookie Cruncher
- Encoder/Decoder
- HTTP Editor
- HP Support Tool
- Log Viewer
- Policy Manager
- Regular Expression Editor
- Server Analyzer
- Server Profiler
- SQL Injector
- Support Channel Tool
- SWFScan
- Web Brute
- Web Discovery
- Web Form Editor
- Web Fuzzer
- Web Macro Recorder (session-based)

- Web Macro Recorder (event-based)
- Web Macro Recorder (TruClient)
- Web Proxy
- Web Services Test Designer

Release 9.30 Features

New Features

WebInspect Enterprise Integration

WebInspect Enterprise is a component of HP Fortify Software Security Center that supports managing your dynamic scanning program. The vulnerabilities discovered during dynamic testing can now be integrated into Software Security Center to provide a central place to manage vulnerabilities detected by both static and dynamic scanning. WebInspect Enterprise enables you to:

- Conduct a large number of automated security scans using any number of sensors to scan Web applications and Web services.
- Manage large or small deployments of HP scanners across your organization controlling product updates, scan policies, scan permissions, tools usage and scan results all centrally from the WebInspect Enterprise console.
- Track, manage and detect new and existing Web applications and monitor all activity associated with them.
- Independently schedule scans and blackout periods, manually launch scans, and update repository information by using HP scanners or the WebInspect Enterprise console.
- Limit exposure to enterprise-sensitive components and data by using centrally defined roles for users.
- Obtain an accurate snapshot of the organization's risk and policy compliance through a centralized database of scan results and trend analysis.
- Facilitate integration with third-party products and deployment of customized Web-based front ends using the Web Services application programming interface (API).

From within WebInspect, you can manage your scans, vulnerabilities and settings and the send the results to WebInspect Enterprise and Software Security Center.

Run a Scan in WebInspect Enterprise

This feature is designed for users who prefer to configure a scan in WebInspect rather than the Assessment Management Platform (AMP) or WebInspect Enterprise. You can modify the settings and run the scan in WebInspect, repeating the process until you achieve what you believe to be the optimal settings. You can then send the open scan's settings to AMP or WebInspect Enterprise, which creates a scan request and places it in the scan queue for the next available sensor.

How?

- 1 Open a Scan.

- 2 Click the **Scan** menu and select **Run in AMP** or **Run in WebInspect Enterprise**.
- 3 Provide a Name, Project and Project Version.
- 4 Click **OK**.

Publish a Scan to Software Security Center

You can transmit scan data from WebInspect to Software Security Center, via WebInspect Enterprise.

How?

To choose from a list of scans:

- 1 Click the **Enterprise Server** menu and select **Publish Scan**.
- 2 Select a scan and select a Project and Project Version.
- 3 Click **Next**.
- 4 Review and select the type of publish.
- 5 Click **Publish**.

To publish from an open scan:

- 1 On the toolbar, click **Synchronize**.
- 2 Select a Project and Project Version.
- 3 Click **OK**.
- 4 On the toolbar, click **Publish**.
- 5 Review and select the type of publish.
- 6 Click **Publish**.

Allow Users to Modify the Status of Vulnerabilities Prior to Publishing

WebInspect Enterprise maintains a history of all vulnerabilities for a particular SSC project version. After WebInspect conducts a scan and you manually synchronize with WebInspect Enterprise to obtain that history, WebInspect compares vulnerabilities in the scan with those in the history, and then assigns a status to each vulnerability. This status can be modified by the user.

How?

- 1 After synchronizing with Software Security Center, right-click a vulnerability and click **Modify Pending Status**.
- 2 Change the status and click **OK**.

New 'Not Found' Tab

This tab is used when synchronizing vulnerability data and uploading to Software Security Center. It lists those vulnerabilities that were detected by a previous scan in a specific project version, but were not detected by the current scan. These vulnerabilities are not included in counts on the dashboard and are not represented in the site or sequence view of the navigation pane.

Support for IBM WebSphere Portal

A template is now available that optimizes settings for IBM WebSphere Portal applications.

How?

Start a scan. When the Scan Wizard displays the Coverage and Thoroughness step, select the **Framework** option and then select **WebSphere Portal** from the dropdown.

Integration of the HP Fortify Seven Pernicious Kingdoms Taxonomy

Seven Pernicious Kingdoms is a taxonomy of software security errors identified by HP Fortify Software Security Research. HP WebInspect now supports the categorization of results by vulnerability Kingdom. The primary goal of the taxonomy is to organize security errors in ways that help software developers understand both their security relevance and program context. By better understanding how systems fail, developers can better analyze systems they create, more readily identify and address security problems when they see them, and generally avoid repeating the same mistakes in the future. For more information, visit <http://www.hpenterprisesecurity.com/vulncat/en/vulncat/index.html>

How?

The Kingdom can be displayed on the **Vulnerabilities**, **Best Practices**, and **Information** tabs by adding this column to the view.

When creating certain reports, you can include the Kingdom by selecting the **Classification** check box.

Enhancements

Support for Security Questions with TruClient Macros

Many companies are enhancing their web form authentication process by adding rotating questions that users must answer each time they log in. For example, the first time you log into a particular site you may be asked "What year did you graduate high school?" and then the next time you try to log in you may be asked "What is your father's birthday?" The TruClient Macro recorder now supports these scenarios by allowing you to select the dynamic element that presents the rotating questions and entering the possible questions and answer combinations. Then whenever the macro is played the location on the page that changes the question is cross referenced to the list of question and answers and the answer is substituted.

Automated Detection of Logout Condition with TruClient Macro Recorder

To help ease recording of successful login macros, the TruClient macro recorder's workflow for recording login macros now includes a step that attempts to automatically identify logout conditions. When the TruClient macro recorder succeeds using its automatic detection methods, the detected logout condition will be added to the recorded macro on user's behalf. If TruClient macro recorder is unable to automatically detect logout condition, the macro recording workflow will prompt the user to pick an element on a page to represent the logout condition.

Easier Proxy Configuration in TruClient Macro Recorder

Configuring the TruClient macro recorder to connect through a proxy has been extended to support additional configuration options. You can now use proxy settings defined in external browsers, such as Internet Explorer or Firefox.

How?

- 1 Define a proxy in Internet Explorer or Firefox.
- 2 Click **General Settings** from within TruClient and select **Proxy Settings**.
- 3 Select **Browser Proxy** and choose either Internet Explorer or Firefox.
- 4 Click **Close**.

TruClient Flash Sites

Previously, when creating a macro for a site containing plug-ins (e.g. Flash), the TruClient browser would display a dialog indicating the plug-in is not installed and will give you the option to install it. This is because plug-ins are not supported in the TruClient browser. This dialog may interfere when macro playback occurs as any action on the dialog is not recorded. Now the dialog no longer appears.

TruClient Clear Passwords

TruClient macros are saved in a plain-text XML format. Unfortunately, this means that any sensitive information (e.g. login credentials) will be in plain text as well. The option to encrypt the entire macro is now available via the Encryption tab in the settings dialog. This option is on by default.

False Positive Improvements

The have been massive improvements in false positive detection by search attacks. These are attacks where WebInspect determines if the server contains a file exists that was not found by the crawler (back-up files, hidden source code, admin directories, framework-specific vulnerabilities, etc.). These tend to flag false positives on sites that return a Status 200 response for any request. This behavior is commonly found in CMS and portal type applications.

Secure Sockets Layer (SSL) Analyzer

WebInspect can now detect if your server is configured to use weak SSL ciphers and weak SSL protocols. SSL is the standard security technology for establishing an encrypted link between a web server and a browser. Using a weak encryption scheme may allow an attacker to view and modify data.

For a detailed list of the defects that were fixed, see the Release Notes.

Release 9.20 Features

New Features

Custom Parameters for URL/REST Rewriting Detection

Many dynamic sites use URL rewriting, thus creating variable elements in the URL. A RESTful web service can also contain parameter names and variable values. Therefore, when WebInspect scans a page, it must be able to determine which elements are variable so that its attack agents can thoroughly check for vulnerabilities. To enable this, you can use the Custom Parameters rule creator to define rules that identify these elements. You can also import them from common configuration files, such as a Web Application Description Language (WADL) file. In addition to the rules you define, WebInspect will also automatically identify custom parameters and suggest them as recommendations.

How?

To define your own custom rules, click the **Edit** menu and select **Default Settings** or **Current Settings**. Then in the Scan Settings category, select **Custom Parameters**.

Scan Comparison

The iterative process of scanning, fixing vulnerabilities and rescanning lends itself to verifying fixed vulnerabilities between scans. The Scan Comparison feature provides a visual representation of vulnerability differences between two scans. It displays the vulnerabilities that are shared by the two scans, as well as any vulnerabilities that are unique to one scan or the other. The information is presented as an interactive dashboard and the common vulnerability view. Use this feature to verify developer fixes, check on scan health, determine new vulnerabilities, investigate issues, compare authorization access, and compare two instances of the same site (such as Production vs. Development).

How?

Scan Comparison can be accessed from the Manage Scans page: select two scans and click **Compare**. It can also be accessed from within an open scan: click **Compare** and select a second scan from the provided list.

Recommendations

As mentioned above, new recommendations are made for custom parameters when URL Rewriting is detected.

Retest Vulnerabilities (Retest in Bulk)

As an extension to the single vulnerability retest functionality introduced in WebInspect 9.00, you can now quickly retest all vulnerabilities detected in a scan. This enables you to determine if a vulnerability still exists without having to conduct an entirely new scan, thus reducing scan time and improving accuracy.

How?

With a scan open on a tab in (or after selecting a scan on the Manage Scans pane), click the **Rescan** drop-down arrow and select **Retest Vulnerabilities**.

TruClient for Login Macros

WebInspect has incorporated HP's TruClient technology as its primary log-in Web macro recording/play-back technology. Capturing and replaying HTTP traffic in order to log into applications is difficult on modern Web applications. However, with TruClient technology, user interactions (not traffic) are captured making log-in playback much more accurate and compatible with sophisticated Web applications.

How?

Start a new scan. When the Scan Wizard displays the Authentication and Connectivity step, select **Site Authentication** and click **Record**. Alternatively, click the WebInspect **Tools** menu and select **Web Macro Recorder**. Note that WebInspect now accommodates three different macro recorders; visit Application Settings - General to select which recorder is launched by default when creating a macro.

Software Security Center Integration

WebInspect 9.2 enables users to export scan data to an HP Fortify Software Security Center, enabling a holistic, centralized view of all Web site security vulnerabilities within one dashboard.

How?

From an open scan, select **File → Export → Scan to Software Security Center** or from the Manage Scans pane, click the **Export** drop-down arrow and select **Export Scan to Software Security Center**.

Common Weakness Enumeration

Common Weakness Enumeration (CWE) is an industry-standard vulnerability classification system from the MITRE Corporation that provides a unified, measurable set of software weaknesses. WebInspect 9.2 incorporates the following support for CWE:

- CWE IDs are displayed in the summary for each vulnerability detected.
- CWE IDs can be displayed, sorted by, and filtered by on the **Vulnerability** tab of the Summary pane.
- CWE IDs can be used to search for appropriate checks in the policy editor.
- CWE IDs are displayed in the Vulnerability Summary and Vulnerability (Legacy) reports.

How?

The CWE ID can be displayed on the **Vulnerability** tab of the Summary pane by adding this column to the view. When creating certain reports, you can include the CWE ID by selecting the **CWE** check box.

Enhancements

Installer/Uninstaller

The following options have been added in the installer/uninstaller:

- Launch WebInspect at the end of successful WebInspect installation.
- Repair a WebInspect installation.
- Return a license when uninstalling WebInspect.
- Remove the product completely, including all program and user data.

How?

Options have been added to the installer and uninstaller.

Supportability

The HP Support Tool provides a quick and simple method for securely uploading files or data that may help HP support personnel analyze and resolve problems you may encounter while using the product. The tool also provides additional tools to fix some common problems. Only use the HP Support Tool as instructed by your HP support personnel.

How?

Click **Help** → **Support** → **Support Tool**.

Web Services Security

Integration of the WS-Security Configuration user interface adds support for Windows Communication Foundation (WCF) and more advanced WS security configurations such as X509 certificates.

How?

From the **WS-Security** tab, within the Web Service Test Designer, click the **Security Tokens** drop-down and select the type of token.

Vulnerability Tab - Export to Excel

This feature allows users to easily export the list of selected vulnerabilities to a comma separated values (CSV) file and then open it in Microsoft Office Excel.

How?

On the **Vulnerabilities** tab of the Summary pane, right-click one or more vulnerabilities, select **Export** on the shortcut menu, and click either **Selected Item(s) to CSV** or **All items to CSV**.

Release 9.10 Features

New Features

WebInspect Real-Time

WebInspect Real-Time features are achieved through integration with HP SecurityScope, an agent that is installed on the target web server. It detects when WebInspect scans the target and provides application information that WebInspect otherwise could not obtain. See below for examples of the type of information that WebInspect can receive from SecurityScope and how it dramatically improves the quality of your scan, reduces the time required to validate your vulnerabilities, and gives developers the information that allows them to fix the vulnerabilities quickly.

Increasing the Attack Surface

Because SecurityScope resides on the target server, it can access the entire breadth of the web application and direct WebInspect to parts of the application that WebInspect may not find through normal crawling and probing. Using this method increases coverage of the application and allows WebInspect to uncover more vulnerabilities.

For example, suppose WebInspect encounters page AddUser.jsp that contains a web form having an input called “department code.” In this case, if you submit the form with a value of 4276, then the user is redirected to HRDepartment.jsp. WebInspect would never discover the HRDepartment.jsp page unless it submitted the web form with the specific value of 4276. Since SecurityScope resides on the server, it informs WebInspect that HRDepartment.jsp exists. In this hypothetical example, WebInspect might then access the page, crawl it for additional links, and discover that a resource named HRAdmin.jsp contains a cross-site scripting vulnerability. In this case, the vulnerability would never have been found with a default configuration of WebInspect.

Stack Trace and Trigger Information

SecurityScope also records additional information about the internal behaviors of the target application during a WebInspect scan and then sends the information to WebInspect for documentation of the vulnerability. This information can be instrumental in understanding the security consequences of a vulnerability. In the case of a SQL injection vulnerability, WebInspect displays the actual SQL statement, as executed on the database server, on the “Stack Traces” option of the Session Info panel. The Stack Trace and Trigger information can optionally be included in the content of the Vulnerability Summary report.

Confirmation of Vulnerabilities

When WebInspect attacks a web application, it sometimes becomes difficult to determine if the attack was successful. In many cases, the application has to respond significantly differently when an attack succeeds versus when the attack fails, or possibly discloses an error message that WebInspect can recognize. With SecurityScope available, the determination of a successful attack no longer depends on the web application reacting differently. For example, in the case of a blind SQL injection attack, SecurityScope can recognize when WebInspect attempts to maliciously attack the database and can determine that the attack was able to access the database without being filtered. It can then inform WebInspect that the attack was successful, even if the web application externally behaved no differently. Subsequently WebInspect can mark that the attack was successful and that it was confirmed by SecurityScope.

Server Identification

Typically WebInspect attempts to fingerprint the web application when a scan begins so that it can customize the attacks that are sent. Sometimes it is difficult to identify the technology used by the application because developers often take steps to hide what platform and application server technology are in use. SecurityScope can easily pass this information to WebInspect, allowing it to tailor its attacks.

Suggestion of Duplicate Vulnerabilities

SecurityScope can determine the vulnerable line of code and the flow that an attack took to reach the vulnerable location. SecurityScope provides this information to WebInspect, which can determine that multiple successful attacks (of the same type) have the same flow and vulnerable location and are therefore most likely duplicate vulnerabilities. The **Vulnerability** tab of the Summary pane groups all the vulnerabilities by duplication, enabling users to review the suspected duplicates and to mark them as ignored (if so desired).

Recommendations - Web Forms

Providing application data as part of configuring a scan is one of the most important steps in getting a high-quality scan of your web application. For example, suppose an application has a page containing a web form with a field for “Employee ID,” and the page cannot be submitted without a valid employee identification number. WebInspect will not be able to submit the form unless someone provides an appropriate value. As the scan progresses, a new recommendations module called “Form Values” will inform you of all the locations where a web form was encountered, but specific values were not provided by the WebInspect user. Once you learn the location, you can easily provide the correct values in the “Form Values” module and rescan the application.

Better Exclusion Management

To obtain a high-quality scan, many users configure WebInspect to exclude certain parts of an application. For example, you may want to exclude the “change password” page from the scan because attacking this page will likely cause the scan to lose session state. In WebInspect 9.10, creating exclusions has become much more flexible by allowing you to create specific rules using multiple criteria for identifying specific pages you would like to exclude from the scan. For example, you could create an exclusion rule such as: URL=Master.jsp and QueryParameter=ChangePassword.

Another key improvement for exclusions is the ability to test the exclusion rule before using it. Misconfiguring an exclusion rule could have drastic impacts on the quality of a scan, so it is important to make sure that the exclusion rule applies only to the intended pages. With WebInspect 9.10, once you create an exclusion rule, you can simply click **Test** to see which pages would have been excluded if that rule had been in effect before the scan began.

Manual Locations

In WebInspect 9.10, you can now manually add locations to the scan that were not found by crawling and probing. This enables you to include vulnerabilities for locations in your WebInspect scan from manual penetration tests, which ultimately can be sent to your defect management system or possibly be included in a single report (both WebInspect and manual results).

Enhancements

Flexible Vulnerability Grid

The **Vulnerability** tab in the Summary pane (the section at the bottom of the scan view that lists all the vulnerabilities) has been enhanced to support grouping and filtering of results. Simply drag column names to the group box to collapse the vulnerability results by different criteria such as vulnerability type, location, or severity. In addition, you can filter by column names in the grid (such as path, severity, parameter, etc.), and you can filter the results based on the contents of the vulnerability. For example, you could filter vulnerabilities that had a response code of 500 or you could filter all the vulnerabilities that had the text “money” in the response. The filtering capability is extremely flexible and allows you to mine your vulnerability results by almost any type of criteria.

Simpler License Model

Previously, connecting WebInspect to the Assessment Management Platform (AMP) required a specific type of license, either an AMP “Fixed” license or a “Roaming” license. Now WebInspect 9.10 with a “Stand-Alone” license can connect to AMP and upload or download scan results.

RPC-Encoded Web Services

Previously, the Web Service Test Designer could not support web services that were RPC-encoded. These types of web services are now supported, although they require manual configuration. The **Schema Fields** tab of the Web Service Test Designer 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.

Release 9.0 Features

New Features

Vulnerability Review with Retest

Understanding the results of an automated security scan is the most time-consuming part of a security audit. Typically the process includes systematically reviewing each reported vulnerability, determining the location of the vulnerability and the path taken by the automated scanner to find the vulnerability, and ultimately attempting to manually reproduce the problem. This can sometimes be challenging for even the most seasoned security professional. WebInspect 9.0 makes huge strides to simplify and streamline this process with the new Vulnerability Review feature.

Reproduction Steps

Simply right-click a vulnerability and select **Review Vulnerability** from the shortcut menu to begin using the new feature. The Vulnerability Review screen presents the path to the selected location (at the bottom of the screen) along with an explanation for the route. For example, if location /AddUsers is vulnerable to cross-site scripting, then the path to the

vulnerability may display a series of locations and reasons such as: /Login - Macro, /Home - Hyperlink, /Users - JavaScript, /AddUsers - Hyperlink. The path the scanner took to the vulnerable location becomes a powerful tool in describing the steps required to reproduce the vulnerability.

Check It Again

Another innovative part of the Vulnerability Review is the new Retest capability. When you click the **Retest** button, WebInspect resubmits the entire vulnerability path to the server, compares each result to the original response, and displays the percentage of retest responses that match the original. This indicates whether the path to the vulnerability was accurately reproduced. Each HTTP request and response for the original and the retest can be compared side by side, instantly revealing any significant variations that would impact the exploitability of the vulnerability. Once the item has been confirmed as a vulnerability, you can submit the defect to either HP Quality Center (ALM) or IBM Rational ClearQuest.

The retest feature is an extremely powerful tool for confirming that developers have fixed a specific vulnerability without having to conduct an entirely new scan. This functionality combined with superior dynamic analysis technology makes WebInspect the superior Web application security product on the market.

New Login Macro Recorder

New Macro Recording Technology

WebInspect now has a new, more accurate method to record login macros. Unlike the old method, which involved replaying HTTP requests, the new tool records user interactions (events) with the Web browser. By interacting with the browser during macro playback, it uses the presentation logic of the Web application to create new, fresh HTTP requests (as opposed to reissuing stale requests that were saved when the macro was recorded).

Enhanced Logout and Confirmation Elements

A logout element is a specific page (or an item that appears on a page) only when the user is logged out of the application. A confirmation element is a specific page (or an item that appears on a page) only when you are logged in. This combination of easily selectable elements ensures that macro playback is nearly infallible.

Clearer Visibility to Success

Once you record a macro, you can replay it to observe each individual user action and browser response, allowing you to determine immediately if the macro will function properly during a Web application scan.

Support for Security Questions

Many companies are enhancing their Web form authentication process by adding rotating questions that users must answer each time they log in. For example, the first time you log into a particular site you may be asked "What year did you graduate high school?" and then the next time you try to log in you may be asked "What is your father's birthday?" The new Login Macro recorder supports dynamic challenge/response authentication by allowing you to specify multiple question-and-answer combinations.

[Redirect to a Different Host](#)

In previous versions, if a hostname for a Web application changed, then the login macro would need to be rerecorded even if there was no change to the content or logic of the Web application. For example, if the hostname for “dev.mycompany.com” changed to “qa.mycompany.com” once it was handed off into the quality phase, then the login macro would normally need to be rerecorded because it was tied to the original location of “dev.mycompany.com.” In the event-based Web Macro Recorder, if you select the setting “Enable URL Replacement,” WebInspect will replace the starting location recorded in the macro with the hostname specified as the Start URL in the scan wizard.

[Improved Web Services Support](#)

[Support for Complex Data Types](#)

WebInspect v9 includes major improvements for interpreting modern WSDLs that incorporate complex data types, recursive types, and other advanced WSDL entities. As part of this effort, WebInspect now includes a new Web Services Security Test Designer tool that renders advanced WSDLs and enables you to specify appropriate application data for your Web service security test. The ability to submit appropriate application data is a critical step in enabling WebInspect to properly exercise and deliver attacks to your Web services.

[WS-Security \(WSS\) Improvements](#)

Supporting all of WS-Security is an enormous feat. As WebInspect marches toward that ultimate goal, this release includes support for User ID and Password credentials at the application layer as well as support for transport layer security (SSL/HTTPS) for Web services.

[New Attacks](#)

Now that WebInspect can interact with a much wider range of Web service deployments, the new Web service infrastructure has been integrated with the existing Smart Engine framework already used as part of Web site scanning. This means that the Local File Include (LFI) and SQL Injection engines can deliver its attacks via Web services, and subsequent future smart engines will easily plug in and enhance both Web site and Web service scanning.

[Better Control](#)

The new Web Services Test Designer (which has replaced the SOAP Editor tool) allows you to control which methods and parameters are sent and/or attacked as part of the Web service scan.

[Web Services Auditing Automatically](#)

If any Web service traffic is detected during a Web site scan, the associated Web service is automatically audited as part of the Web site scan. This enables you immediately to get coverage against the larger attack surface of the Web application even if you may not know that Web services are present. Although this functionality is helpful, a full Web service scan is recommended in addition to any Web service scanning that occurs during a Web site scan.

Reuse Identified False Positives across Scans

As stated previously, understanding the results of an automated security scan is the most time-consuming part of a security audit, and marking any misreported vulnerabilities as false positives is an important step in pruning the results. In previous versions, once a false positive was identified in a scan, that information was not available to future scans, requiring the user to repeatedly mark the same vulnerability as a false positive. In WebInspect 9.0 this problem has been solved. When you start a Web site assessment, you can optionally choose an existing scan from which to “copy” false positives. As the scan runs, WebInspect compares each detected vulnerability to the copied false positives and automatically marks matched vulnerabilities as false positive. The vulnerability chart on the scan dashboard indicates this process by incrementing the false positive count as the scan progresses.

You can also select to copy false positives from an existing scan after both scans have been completed (or paused). This is especially helpful in cases where you may have conducted a multi-hour scan but forgot to select the Import False Positive option in the scan wizard. Simply access the False Positive section in the Scan Info panel to import false positives.

Post Scan Analysis / Recommendations

Post scan analysis provides users with recommendations on how to optimize their scans by analyzing specific conditions in the scan results. Post Scan Analysis is a collection of plug-ins that allow for future expansion with additional modules. Once a recommendation is made, the user can review the recommendation and either accept it (by changing the configuration change) or ignore it.

- Authentication - Network authentication challenges were received during the scan, indicating credentials are either missing or incorrect. This recommendation becomes very useful in indicating parts of the site you may not know exist (such as a probe to the /admin directory that results with an NTLM challenge).
- Web Macro - Your login macro is excessively replaying, most likely indicating a problem with the trigger condition or possibly a problem with the larger scan that is causing repeated logouts to occur.
- File Not Found - For an automated scanner, it is sometimes difficult to determine if a Web application is delivering actual content versus an HTTP 200 response code with an interesting message saying that a page does not exist. WebInspect has various methods to determine this situation, but if it is not correctly identified there will be dramatic errors in the resulting scan; primarily every probe will flag as a vulnerability when it is actually a false positive. The File Not Found recommendation module examines the total number of probes that positively flagged against the total size of the scan results and determines if a problem with File Not Found detection is likely.
- Web Service Detection - While the Web application is crawled or audited, it WebInspect detects a call to a Web service, then this recommendation alerts you that the Web service exists and reveals the location of the Web service. The recommendation module also allows you to export the captured SOAP messages so they can be used during a Web service scan. This becomes a very useful feature because it allows you to use actual application data for your Web service scan without requiring you to input it manually.

Post Scan Analysis will occur when a scan is complete or paused. As a best practice, consider pausing your scan after a period of time to determine if there are any high-value recommendations made that would warrant restarting your Web site scan.

New Blind SQL Injection Engine (BSQLI)

A new method for detecting blind SQL injection has been developed. Known as Timing-Based, it sends SQL injection attacks aimed at slowing the performance of the database server for a specific period of time. Once the attacks are sent, the engine performs a sophisticated technique for sampling the response time of the Web application to gauge whether the SQL attack succeeded. This new method complements the preexisting inferential testing method.

DOM-Based Cross-Site Scripting (DOM-XSS)

An attacker can execute malicious JavaScript code in the context of the victim's browser by modifying the document object model (DOM) environment. Such modifications can cause the original script using the tainted DOM to behave in an insecure manner. This DOM-based version of cross-site scripting (XSS) differs from the stored and reflected XSS in that the malicious data is never sent to the server. WebInspect 9.0 incorporates an improved cross-site scripting audit engine that now has the ability to detect and report these vulnerabilities.

New Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) is one of the most malicious and more important Web application security vulnerabilities (OWASP Top 10). Typically, this type of vulnerability is exploited when a user interacts with a particular site and that site sends a request to another site using a previously established cookie to perform an action without the knowledge or permission of the user. For example, suppose you log into your banking site (MyBank.com) and then later access the questionable MyPersonalBlog.com site. Hovering your mouse over a button could cause the .onhover method to be called which sends the malicious request `http://mybank.com/TransferAllYourMoney.aspx?from=youraccount&to=myaccount`. Normally, mybank.com would prompt the user to authenticate before allowing access to the transferallyourmoney.aspx page, but because an authentication cookie was already established, the request would succeed. In this case, mybank.com was vulnerable to CSRF.

A new CSRF check is available in all versions of WebInspect 8.0 and above, and can be downloaded via SmartUpdate. It first attempts to identify from context whether the Web application is attempting to protect a particular location (such as Change Password, TransferMoney, etc.). Then the check emulates a real CSRF attack against a protected location to determine it is vulnerable. Most CSRF vulnerabilities found by WebInspect initially are flagged as 'medium' severity vulnerabilities, and users need determine its actual rating in terms of application contexts.

Rescan

Rescan is a handy feature that allows you to easily transition from an open or selected scan into the scan wizard with the original scan settings preloaded. Historically, it might require several modification of the scan configuration to obtain a scan that adequately crawls and audits a Web application; rescan streamlines this process. The rescan functionality is available in two areas: the Rescan button on the scan toolbar and the Rescan button (and shortcut menu) for a selected scan on the *Manage Scans* pane.

Attachments and Screenshots

Screenshots are an integral way to describe the steps to reproduce a vulnerability as well as convince skeptics that a vulnerability exists and needs to be addressed. In WebInspect 9.0, you can upload screenshots to your scan results and associate them with a specific vulnerability. The option to "Copy from Clipboard" is a handy shortcut to directly add screen

captures without having to first save the image to disk. You can add a screenshot by selecting the vulnerability on the Vulnerability tab in the Summary pane and then selecting the Attachments option.

Once a screenshot has been added to a vulnerability, it will then be optionally included in any vulnerability report that is generated. In addition, the screenshots will be included as attachments when the associated vulnerability is sent to HP Quality Center (ALM) or IBM Clear Quest defect management systems. The attached screenshots will also be included with the scan results when uploading scans from WebInspect to AMP.

AMP Integration - Support for Assessments

AMP v9.00 introduces the new concept of assessments. An assessment is a virtual workspace for scans and vulnerability information, allowing you to bring together all the results of your Web application security testing into one centralized location.

WebInspect v9.00 supports AMP's new assessment functionality by allowing users to choose the assessment into which the scan should be uploaded. If necessary, WebInspect users can also create an assessment in AMP before uploading the scan.

Detachable Concurrent Licenses

WebInspect concurrent licenses can be detached from the License and Infrastructure Manager (LIM) v2.0. WebInspect 9.0 allows users with concurrent licenses to check out (lease) a license for a specified period of time. The license then is not available for use by other WebInspect concurrent license users until it is either returned or the lease expiration date has occurred.

Enhancements in Version 9.00

WebInspect version 9.00 includes the following enhancements:

Export of Vulnerabilities Only

In previous versions of WebInspect, if you wanted a list of vulnerabilities in a parseable format, the only option was to export the entire scan to an xml file. This very large file would contain all the information necessary to reconstitute every location found in the scan (including non-vulnerable locations). WebInspect v9.00 has been enhanced to allow users to export only the vulnerabilities, which is a much smaller subset of the scan. Simply open the scan that contains the vulnerabilities you would like to export and navigate to the Export Details screen (**Export > Export Details**) and select **Vulnerabilities**.

Scheduler Improvements

Scheduled scans that are currently running can now be stopped using the Manage Schedule interface on the WebInspect Start Page tab. Before this release, scheduled scans ran until completion. Simply select an actively running scan on the “Manage Schedule” section and click the **Stop** button. A stopped scan can be opened, analyzed, and resumed if desired.

Traffic Monitor Improvements

It can be very difficult to understand exactly why the automated scanner is sending certain HTTP requests to a Web application or what HTTP requests relate to which engines or checks. With these challenges in mind, the traffic monitor was enhanced in WebInspect v9.00 to include more diagnostic information; specifically columns for Engine, Check ID, Source, etc. If you would like to know all the requests made because of a specific check, simply sort by the CheckID column.

Enabling the traffic monitor will lengthen the time required to conduct scans and will increase the use of disk space.

Recovering Ignored Vulnerabilities and Deleted Sessions

Have you ever accidentally marked a vulnerability as ignored or deleted a session only to regret it a moment later? Previously if you mistakenly marked an item as ignored or deleted, you were required to call HP Support to reverse the change. Now, you can now reverse the change yourself. Click the Dashboard option on the Scan Info panel to view statistics related to the scan. Then simply click the hyperlink representing the number of deleted items and choose which vulnerabilities and/or sessions you would like to restore.

Integration with the SWFScan Flash Analyzer

The free SWFScan tool released by HP is now integrated with WebInspect. If a Flash file is detected during a scan and appears on the site tree in the navigation area, simply right-click the item and launch SWFScan. From there you can explore the file's methods and source code to manually verify any vulnerabilities.

Application Settings

Reset 'Do Not Show' Messages

You can now visit the application settings to turn on all the "Do Not Show" messages. Once that has been done, WebInspect will once again prompt you with these informative messages.

Database Scan Configuration

The database configuration for scans has been moved from the advanced scan settings configuration to application settings. Feedback from our customers indicates that they tend to create a single database for all their scans and that multiple remote SQL databases are rarely deployed. Therefore, we have simplified database configuration by making it a product-wide option rather than an option that can be configured per scan.

New Compliance Template

A new PCI DSS 2.0 compliance template has been created and included in this release.

Easier Installation

WebInspect v9.00 bundles the following software programs into the installation package so these prerequisites can optionally be deployed as part of the WebInspect installation process.

- Windows Installer 4.5 (Prerequisite for SQL Server Express 2008 R2)

- Microsoft SQL Server Express 2008 R2 (not needed if you use SQL Server Standard Edition)
- Microsoft .NET Framework 3.5 Service Pack 1

Release 8.10 Features

The following features are included in HP WebInspect release 8.10.

New Scan Wizards

A new scan wizard has been added that is easier to use and provides additional ways to scan your application.

- **List Driven Scan:** Allows you to scan a list of URLs instead of just starting from a single location. This helps ensure better coverage of your target web application. This list can be provided by creating a text file containing the list of URLs or could be obtained by running the FilesToURLs.exe utility on your target site, which will create an XML file containing a list of URLs to investigate from your source code. For more information, see [FilesToURLs Utility](#) on page 162.
- **Workflow-Driven Scan:** Allows you to record a “path” through your application and assess that specific workflow. This is useful when assessing web sites that require a very specific set of steps in order to navigate through the application.
- **Flexible Scan Modes:** With the new scan wizard, you can now separately choose your activity from your application target. For example, you can choose whether to audit a workflow macro (in essence auditing a set of steps) or crawl and audit a workflow macro (use the macro to improve the automated crawler), or many other permutations.
- **Prepared Settings for Target Frameworks and Applications:** HP Engineers have developed a set of options that are optimized for scanning Web sites created with Oracle ADF Faces.
- **Easier Control of Crawling your Application:** With the new scan wizard you can decide if you want a very thorough crawl, quick crawl, or something in between. This allows you to balance between scan time and depth of the crawl.

If you prefer, you can still use the classic scan wizards by selecting an option in the Application settings.

Support for Windows 7 and Windows Server 2008

WebInspect now supports Microsoft Windows 7 and Microsoft Windows Server 2008.

Improved Integration with the Application Management Platform (AMP)

- **Settings Transfer to and from AMP:** AMP-enabled WebInspect installations can now create an AMP scan template based on a WebInspect settings file and upload it from WebInspect to AMP. AMP users can also create a WebInspect settings file based on an AMP scan template and download it from AMP to WebInspect.
- **Run Scan in AMP:** AMP-enabled WebInspect installations can request AMP to run a scan. AMP places the request in a queue to be serviced by any available sensor.

- **Support for Organizations and Project Hierarchy:** Beginning with AMP 8.10, sites are assigned to a project and projects are assigned to an organization. When uploading a scan from WebInspect to AMP, users are able to specify the organization, project, and site in which the scan should be placed.

Support for Organization/Project Hierarchy in AMP

Beginning with AMP 8.10, sites are assigned to a project and projects are assigned to an organization. When uploading a scan from WebInspect to AMP, users are able to specify the organization, project, and site in which the scan should be placed.

Smart Response Truncation for Reports

Generated reports can contain very lengthy HTTP request and response messages. To save space and help focus on the pertinent data related to a vulnerability, you can exclude message content that precedes and follows the data that identifies or confirms the vulnerability (identified by red highlighting). To use smart truncation in reports, select **Smart truncate vulnerability text** in the Reports section of the Application settings and then specify the number of characters to retain preceding and following the data that identifies or confirms the vulnerability. See [Smart truncate vulnerability text](#) on page 217 for more information.

Better Phase Identification for Crawl-Only Scans

In previous versions of WebInspect, the audit progress bar on the scan dashboard appeared to indicate audit activity when users performed a crawl-only scan, when in fact WebInspect was conducting a keyword search or path truncation. The dashboard now displays the qualifier “Audit (Passive)” or “Audit (Discovery)” to indicate that actual audits are not being conducted.

Release 8.00 Features

WebInspect is the premier Web application security scan tool designed specifically for today's complex Web applications built on emerging Web 2.0 technologies. This new architecture delivers faster scanning capabilities, broader scan coverage, and the most accurate results of any Web application scanner available.

New Features

The following new features are included in HP WebInspect release 8.00.

Flash Static Analysis

WebInspect can now decompile the latest version of Shockwave Flash (SWF) files and then perform static analysis on the resulting ActionScript 3 code, detecting vulnerabilities such as insecure programming practices, insecure application deployment, Adobe “best practices” violations, and information disclosure.

New Reporting System

WebInspect's new and powerful reporting system facilitates the presentation of analyzed data. Now you can:

- Create reports that are flexible, scalable, and faster using an improved generation workflow.

- Modify standard reports or design your own using our new report designer.
- Extract information from external data sources.
- Customize fonts, colors, and backgrounds with the new Report Style Editor.
- Generate scan reports with a professional, polished appearance.
- Focus analysis on a single session with our new session reports.

[Optional Depth-First Crawler](#)

Depth-first crawling accommodates sites that enforce order-dependent navigation (where you must visit page A before you can visit page B). This method traces the first link on a page to the first link on the referenced page before returning to the original page and tracing the second link. By contrast, breadth-first crawling (which is also available) follows all the links on a page before drilling down to the pages that are linked.

[Java Model View Control \(MVC\) Support](#)

Based on in-depth research by the HP DevInspect for Java team, WebInspect now supports applications built on the Java MVC platform by the use of the depth-first crawler, path-based attacks, and navigational parameters.

[Integration with IBM Rational ClearQuest](#)

You can now send vulnerabilities as defects directly to IBM Rational ClearQuest version 7.

[Support for 64-Bit Vista](#)

You can now run WebInspect on 64-bit Vista operating systems, allowing more memory for both WebInspect and SQL Server.

[Improvements and Enhancements](#)

The following WebInspect features have been improved.

[Script Processing](#)

WebInspect now handles applications that use heavy client-side JavaScript. With the Web 2.0 era and the push for applications to reside more on the client, server-side page-centric applications are slowly diminishing. As applications move to the client, they become a single page that delivers an application almost entirely written in JavaScript, making it very difficult for a scanner to follow links when crawling the application. Crawling is becoming more about tracing code paths through the JavaScript, analyzing how the application changes from the user's perspective, and watching AJAX requests and making attacks to the server accordingly. WebInspect 8.00 delivers breakthroughs in JavaScript technology by tracing and recording code paths as subsessions, which are then audited to reveal vulnerabilities.

[Web Macro Recorder](#)

The Web Macro Recorder is now easier to use and incorporates a new algorithm for determining a logout condition. Once you record the login sequence, the Web Macro Recorder automatically samples the Web site to discover specific keywords that are present when state has been acquired and when it has been lost. This allows the scanner to reacquire state if it inadvertently becomes "logged out." The Web Macro Recorder also now allows you to verify that your macros work as expected before you use them.

Smart Scan Fingerprinting

WebInspect is now more accurate than ever when choosing which checks to use against Web sites. It runs a series of fingerprint requests to determine the server type, version, and platforms supported.

Start Page

The layout, appearance, and general usability of the Start page have been improved. It displays new scan attribute columns in the Manage Scans workspace, which improves scan selection. You can also group scans by scan attributes. The Activity Panel is now collapsible to increase your Manage Scans workspace.

Scan View

Excluded hosts and allowed hosts are distinctly grouped, and the Scan statistics panel has been moved to the right of the dashboard for a better look and feel. The Scan Dashboard has an improved layout featuring a prominent scan status, crawl and audit activity indicators with rolling performance counters, script (JavaScript and VBScript) execution indicator, and a listing of attack engines grouped by attack type.

2 Getting Started

Introduction

This chapter outlines the requirements and procedures for installing WebInspect. It also contains suggestions for preparing your system for audit.

Software Installation

Requirements

Before installing WebInspect, make sure that your system conforms to the following list of supported components. Note that Beta versions of operating systems, service packs, and required third-party components (such as Microsoft SQL Server Express) are not supported.

Supported Operating Systems

- Windows 7 (32-/64-bit) (Recommended)
- Windows XP Professional SP3 (32-bit)
- Windows Vista SP2 (32-/64-bit)

- Windows Server 2003 Standard SP2 (32-bit)
- Windows Server 2008 R2 (64-bit) (Recommended)

Processor

- 1.5 GHz Single-Core (Minimum)
- 2.5 GHz Multi-Core (Recommended)

RAM

- 2 GB (Minimum)
- 4 GB (Recommended)

Hard Disk

- 10 GB (Minimum)
- 100+ GB (Recommended)

Display

- 1024 x 768 (Minimum)
- 1280 x 1024 (Recommended)

Supported Database

- Microsoft SQL Server Express Edition 2008 R2 (10 GB scan database limit)
- Microsoft SQL Server Express Edition 2008 SP2 (4 GB scan database limit)
- Microsoft SQL Server Express Edition 2005 SP4 (Minimum) (4 GB scan database limit)
- Microsoft SQL Server 2008 R2 (Recommended) (No scan database limit)
- Microsoft SQL Server 2008 SP2 (No scan database limit)
- Microsoft SQL Server 2005 SP4 (No scan database limit)

Platform

- Microsoft .NET Framework 3.5 Service Pack 1

Browser

- Internet Explorer 7.0
- Internet Explorer 8.0 (Recommended)
- Mozilla Firefox 3.6 (Proxy Settings Only)

Integrations

- HP Assessment Management Platform (AMP) v8.00, v8.10, v9.00, v9.10, v9.20
- HP Quality Center (QC) v9.2, v10.0
- HP Application Lifecycle Management (ALM) v11.0
- IBM Rational ClearQuest v7.1
- WebInspect Enterprise v9.30

PDF Support

- Adobe Acrobat v10.0 (Recommended)
- Adobe Acrobat v8.0 (Minimum)

Network

- An active Internet connection (for updates)

Notes about Vista/Windows 7 Installations

- Disable User Account Control (UAC).
- Disable themes (recommended).
- When installing, make sure that you run as Administrator.

- On occasion SQL Compact Edition, a prerequisite for SmartUpdate of SecureBase Checks, will not correctly register into the Global Assembly Cache (GAC) on 64-bit platforms (Windows Vista and Windows 7). In order to ensure that registration occurs correctly, consider installing SQL Compact Edition 64-bit edition prior to installing WebInspect.

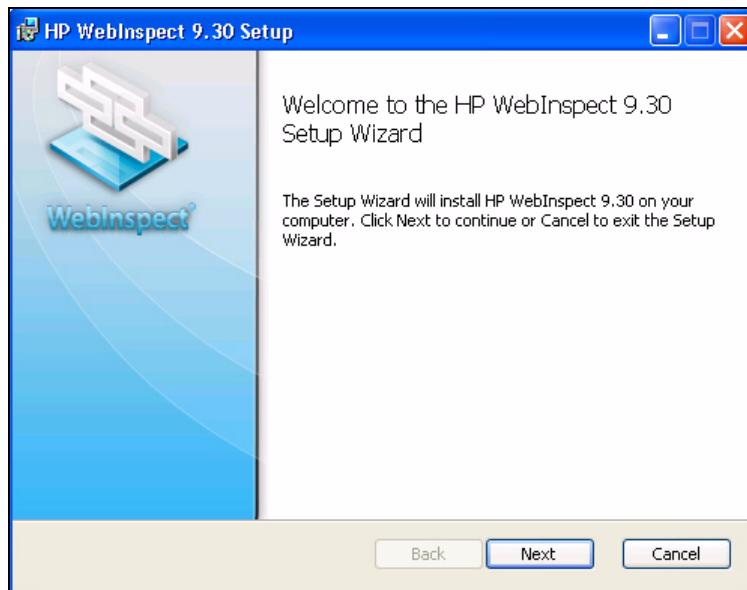
Notes about SQL Server

- When using SQL Server Express Edition, scan data may not exceed that maximum database size for the version of SQL Server Express Edition installed (4GB for 2005/2008 and 10GB for 2008 R2). For larger scans, or to permit sharing of scan data, use SQL Server (full version).
- SQL Express: Install taking all of the defaults, as the product is expecting the default instance name “SQLEXPRESS.” The user may also wish to enable the boxes for “Hide advanced installation options” and “Add current user to database administrators.”
- SQL Server 2005/2008 (full): This may be on the local host or collocated nearby. This option can be configured within the Application Settings: **Edit → Application Settings → Database.**

Installation

Use the following procedure to install WebInspect.

- 1 Start the installation program.
- 2 On the Welcome page, click **Next**.

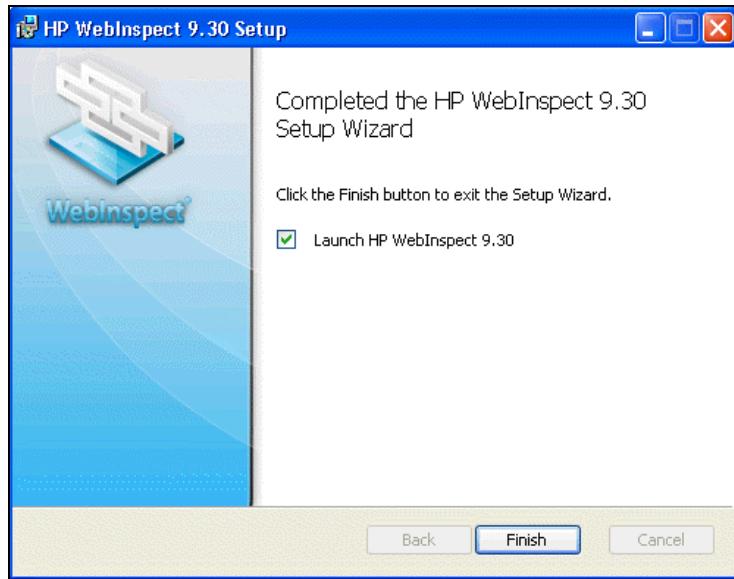


- 3 Review the license agreement. If you accept, select the check box and click **Next**; otherwise click **Cancel**.
- 4 On the *Destination Folder* window, select the folder into which you want to install the software and click **Next**.

The *Sensor Configuration* window appears.

- 5 If you are installing WebInspect as a sensor for the Assessment Management Platform (AMP) or WebInspect Enterprise:

- a Select **Configure WebInspect as a Sensor**.
 - b Enter the URL of the AMP manager or WebInspect Enterprise manager.
 - c In the **Sensor Authentication** group, enter the Windows account credentials for this sensor.
- 6 Click **Next**.
- 7 On the *Ready to Install* window, click **Install**.
- 8 When the process is complete, select **Launch HP WebInspect 9.30** and click **Finish**.

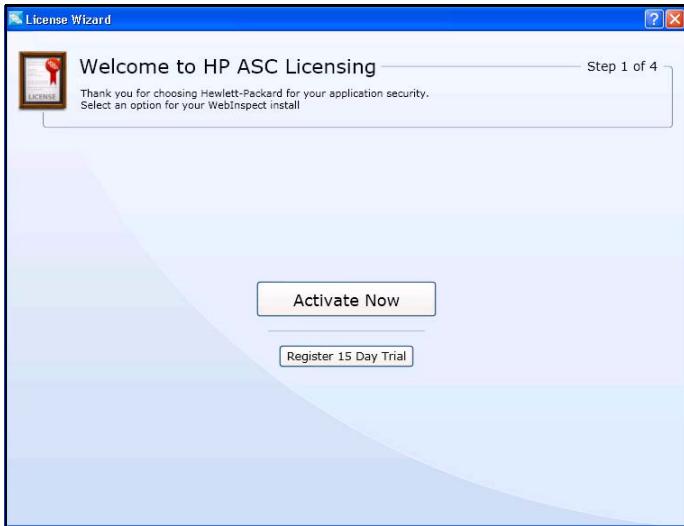


Licensing

The first time you start WebInspect, the program displays the License Wizard, which prompts you to select one of the following options:

- Activate Now

- Register 15-day trial



Trial Registration

Use the following procedure to begin a free 15-day trial of WebInspect.

- 1 On the *WebInspect Product Registration Wizard* window, click **Register 15 Day Trial**.

The wizard displays a window prompting you to enter information about you and your company.

- 2 Enter the requested information.
- 3 If connecting to the Internet through a proxy, click Connection Settings, modify the settings (if necessary) and click **OK**.
- 4 Click **Next**.

The program attempts to contact HP servers, which will send an e-mail message to you containing a 32-character activation token.

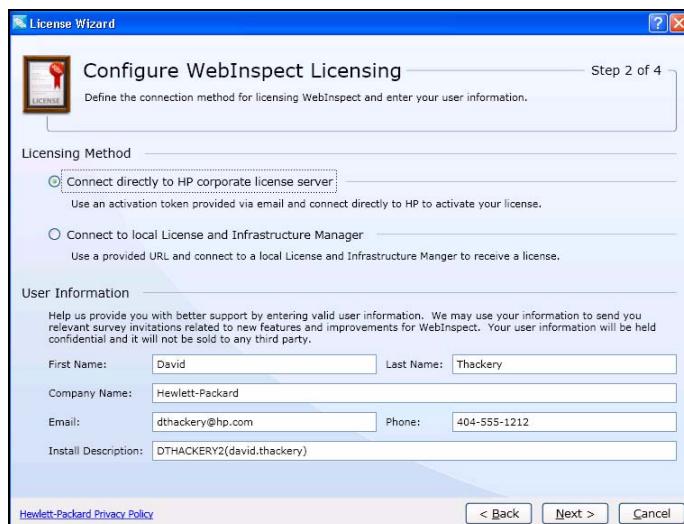
- 5 Click **Finish**.
- 6 When the e-mail arrives, click the WebInspect **Edit** menu and select **Application Settings**.
- 7 On the *Application Settings* window, select **License** from the left pane.
- 8 Enter the 32-digit license token, omitting any hyphens that may appear in the string (or simply copy the token, position your cursor in the first block of the **Activation Token** field, and press **Ctrl + V**).
- 9 (Optional) Enter a description.
- 10 Click **OK**.

Activate Now

Select this option if you have received a token in an e-mail message from HP.

- 1 On the License Wizard window, click **Activate Now**.

The wizard displays the Configure WebInspect Licensing panel.



- 2 In the Licensing Method group, choose one of the following:
 - **Connect directly to HP corporate license server** - Licensing is controlled by a Hewlett-Packard server.
 - **Connect to local HP License and Infrastructure Manager** - Licensing is controlled by your local server running HP License and Infrastructure Manager (LIM) software.

The LIM allows you to manage concurrent licenses for HP software in a manner that best suits your organization's development and testing environment. For example, your company may have HP WebInspect software installed on 25 machines, but holds a concurrent license that permits a maximum of 10 instances to be active at any one time. Using the LIM, you can allocate and deallocate those 10 seats in any way you like, without coordinating or negotiating through HP's central licensing facility.

- 3 Enter the information requested in the User Information group.
- 4 Click **Next**.

If you chose **Connect directly to HP corporate license server**, the License Wizard displays the Named License Activation panel. Go to Connect to HP.

If you chose **Connect to local HP License and Infrastructure Manager**, go to Connect to LIM.

Connect to HP

- 1 In the Activation Token area, enter the 32-digit license token sent to you by e-mail from HP. Omit any hyphens that may appear in the string (or simply copy the token, position your cursor in the first block of the Activation Token field, and press Ctrl + V).
- 2 If the machine you are attempting to license is connected to the Internet, select **Online Activation**.
 - a The default URL of the HP license service is
<https://LicenseService.HPSmartUpdate.com>.
Change this only if directed to do so by HP Support personnel.
 - b If you access the Internet through a proxy, select **Network Proxy** and choose a setting from the **Proxy Profile** list. You must also click **Edit** if you select **Use PAC file** or **Use Explicit Proxy Settings**.
 - **Use PAC File** - Load proxy settings from a Proxy Automatic Configuration (PAC) file. Enter the location of the PAC file in the URL box.
 - **Use Explicit Proxy Settings** - Configure a proxy by entering the requested information:
- 3 If WebInspect is installed on a computer that is not connected to the Internet, select **Offline Activation**. You will create a license request file containing information about that computer and then, using a separate Internet-connected computer, access an HP Web site to transmit the file to an HP server, which will download a license file that you can copy and install on the computer that is not connected to the Internet.
 - a Click the Browse button next to the **License Request File** box.
 - b Select a location where the file will be saved. The name of the request file is formatted as <ProductName>_LicenseReq.xml.
Be sure to save this file on a portable media or at a location that is accessible by a machine that has access to the Internet.
See the on-line Help for additional instructions.
- 4 Click **Next**.
Information pertaining to your installed license appears in the License Details section.
- 5 Click **Finish**.

Connect to LIM

The HP License and Infrastructure Manager (LIM) allows your company to manage concurrent licenses for HP software in a manner that best suits your organization's development and testing environment. Contact your LIM administrator to obtain the information required for completing this procedure.

- 1 In the **URL** box, enter the URL of the License and Infrastructure Manager.
- 2 Enter the name of the license pool and the pool password in the appropriate boxes.
- 3 If authorization is required to access the LIM, select **Network Credentials** and then enter your user name and password.

- 4 If connecting to the LIM through a proxy, select **Network Proxy** and choose a setting from the **Proxy Profile** list. You must also click **Edit** if you select **Use PAC file** or **Use Explicit Proxy Settings**.
 - 5 Click **Next**.
 - 6 On the Complete on-site License Activation panel, select the manner in which you want the License and Infrastructure Manager to handle the license associated with WebInspect.

 - **Connected License** - The computer can run the HP product only when the computer is able to contact the LIM. Each time you start the HP software, the LIM allocates a seat from the license pool to this installation. When you close the software, the seat is released from the computer and allocated back to the pool, allowing another user to consume the license.
 - **Detached License** - The computer can run the HP product anywhere, even when disconnected from your corporate intranet (on which the LIM is normally located), but only until the expiration date you specify. This allows you to take your laptop to a remote site and run the HP software. When you reconnect to the corporate intranet, you can access the Application License settings and reconfigure from Detached to Connected.
 - 7 Click **Next**.
- Information pertaining to your installed license appears in the License Details section.
- 8 Click **Finish**.

License Revocation

If your WebInspect license expires, or if your facility is managing licenses through the HP License and Infrastructure Manager and the administrator releases your license, you will not be able to conduct or schedule scans.

To regain a license if using the HP License and Infrastructure Manager:

- 1 Click the **Edit** menu and select **Application Settings**.
- 2 Click **License**.
- 3 Verify your license data.
- 4 Click **OK**.

If necessary, contact HP Support or your HP License and Infrastructure Manager administrator.

Preparing Your System for Audit

WebInspect is an aggressive Web application analyzer that rigorously inspects your Web site for real and potential security vulnerabilities. This procedure is intrusive to varying degrees. Depending on which WebInspect policy you apply and the options you select, it can affect server and application throughput and efficiency. When using the most aggressive policies, you should perform this analysis in a controlled environment while monitoring your servers.

This section discusses precautions and advisements that you should consider before conducting a vulnerability scan with WebInspect.

Increased Traffic

During an audit of any type, WebInspect submits a large number of HTTP requests, many of which have “invalid” parameters. On slower systems, the volume of requests may degrade the system or deny access to other users. Additionally, if you are using an intrusion detection system, it will identify numerous illegal access attempts.

WebInspect can be configured to send up to 75 concurrent HTTP requests before waiting for an HTTP response to the first request. The default thread count setting for a sequential crawl and audit is 5 for a crawl and 10 for an audit (if using separate requestors). Increasing the thread count will increase the speed of a scan, but might also exhaust your system resources as well as those of the server you are scanning. See Requestor on page 160.

Form Submission

To conduct a thorough scan, WebInspect attempts to identify every page, form, file, and folder that composes your application. If you select the option to submit forms during a crawl of your site, WebInspect will complete and submit all forms it encounters. Although this enables WebInspect to navigate seamlessly through your application, it may also produce the following consequences:

- If, when a user normally submits a form, the application creates and sends e-mails or bulletin board postings (to a product support or sales group, for example), WebInspect will also generate these messages as part of its probe.
- If normal form submission causes records to be added to a database, then forms submitted by WebInspect will create spurious records.

During the audit phase of a scan, WebInspect resubmits forms numerous times, manipulating every possible parameter to reveal problems in the applications. This will greatly increase the number of messages and database records created.

WebInspect submits data extracted from a prepackaged file. If you require specific values (such as user names and passwords), you must create a file with HP’s easy-to-use Web Form Editor and identify that file to WebInspect.

For systems that write records to a back-end server (database, LDAP, etc.) based on forms submitted by clients, some WebInspect users, before auditing their production system, create a backup copy of their database and then reinstall it after the audit is complete. If this is not feasible, you can query your servers after the audit, searching for and deleting records that contain one or more of the form values used by WebInspect.

To prohibit these automated actions from occurring, you should prevent them at the system level so that WebInspect can still fully test the interfaces. For instance, disabling the SMTP service on the Web application may prevent the sending of inadvertent e-mails, but still allow WebInspect to audit the submitted forms. On a mirrored database, it may be possible to disable or remove all triggers to prevent inadvertent actions. However, if you choose not to prevent these automated actions at the system level, then the architect or development team should identify all “off limit” pages in advance, and these URLs should be excluded within the default settings in WebInspect.

E-mail Messages

If your system generates e-mail messages in response to user-submitted forms, you might consider disabling your mail server. Alternatively, you could redirect all e-mails to a queue and then, following the audit, manually review and delete those e-mails that were generated

in response to forms submitted by WebInspect. Another alternative is to disable the **Auto-fill web forms during crawl** setting (see Scan Settings: [Method](#) on page 165), although this could limit the scope of the scan. You could also identify those pages on which e-mails are submitted and specify them as excluded URLs (see [Session Exclusions](#) on page 178); this may also limit the scope of the scan. Note that none of these precautions would prevent e-mail transmissions that are initiated programatically (such as a failed authentication attempt triggering an e-mail to your security center).

Binary Documents

By default, WebInspect is configured to ignore many binary files (images, documents, etc.) that are commonly found in Web applications. These documents cannot be crawled or attacked, so there is no value in auditing them. Bypassing these documents greatly increases the speed of the audit. If there are proprietary documents in use, try to ascertain the extensions of the documents and exclude them within WebInspect's default settings. If, during a crawl, WebInspect becomes extremely slow or even halts, there is a chance that it attempted to download a binary document.

Sensitive Areas

If for any reason you do not want WebInspect to crawl and attack certain directories, you must specify those directories using the Other Exclusion/Rejection Criteria feature of WebInspect settings. See page 165.

Added Files

Finally, WebInspect tests for certain vulnerabilities by attempting to upload files to your server. If your server allows this, WebInspect will record this susceptibility in its scan report and attempt to delete the file. Sometimes, however, the server will not allow a file to be deleted. For this reason, part of your post-scan maintenance should include searching for and deleting files whose name begins with "CreatedByHP".

Updating WebInspect

HP security engineers uncover new vulnerabilities nearly every day. They then develop attack agents to search for these malicious threats, then update our corporate database so that you will always be on the leading edge of Web application security.

To ensure that you have up-to-date information about the WebInspect catalog of vulnerabilities, you can instruct WebInspect to contact our knowledgebase server each time you start the application. If vulnerability or program updates are available, WebInspect informs you and asks if you want to install them.

For complete information about updating WebInspect, see [Smart Update](#) on page 290.

Directory Structure

The following table describes the directories created and used by WebInspect, assuming that the main drive is “C” and the user accepts the default directories suggested by the installation program.

WebInspect Directory Structure

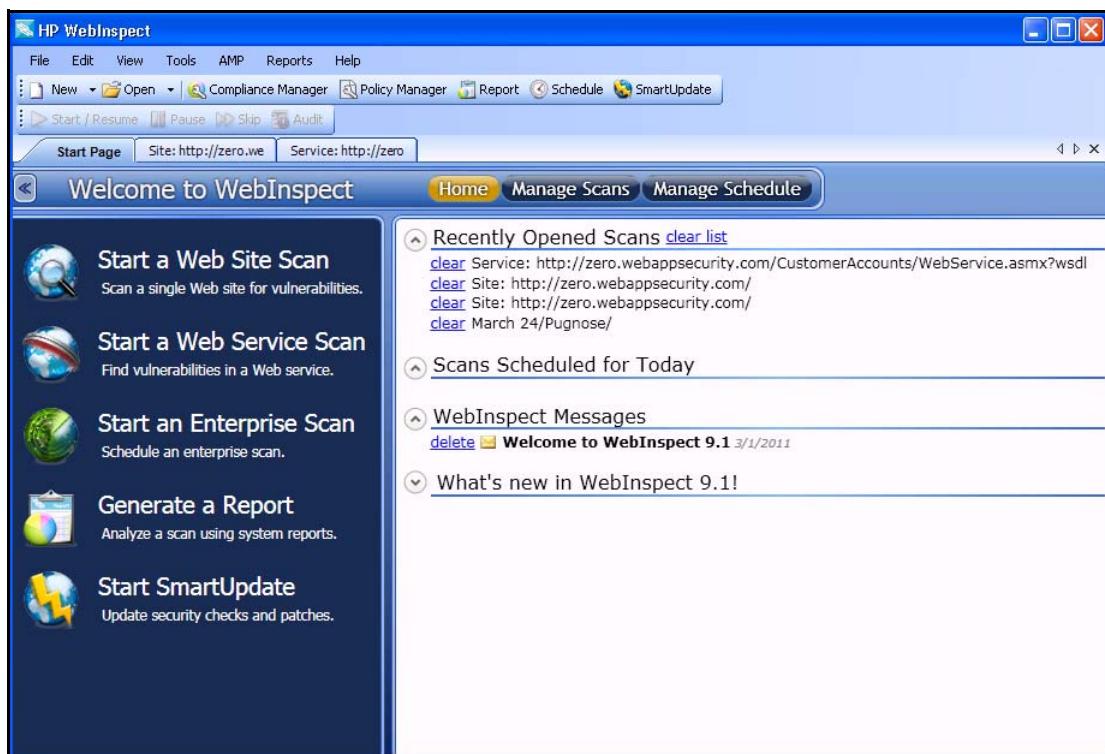
Purpose	Path	Remarks
Installation Directory	C:\Program Files (x86)\HP\HP WebInspect	Can be set by the user during installation. SmartUpdate of full WI version will override everything that exists in this directory.
	<Install Directory>\Compliance Templates	Compliance template directory; can be modified by Smart Update.
	<Install Directory>\Samples	Contains subdirectories for policies, sample scans, settings for HacmeBank.com, a login macro and WSDL file for zero.webappsecurity.com,
Application Data Directory	C:\Documents and Settings\All Users\Application Data\HP\	All data required for WebInspect that are not user-specific.
	<Application Data Directory>\HP WebInspect	Subdirectories include Policies, Schedule, SecureBase database, Server Analyzer, Settings., and Support Channel.
	<Application Data Directory>\Licenses	Licenses activated on the local machine.
	<Application Data Directory>\SmartUpdate	SmartUpdate directory where new patches are downloaded. Security checks are copied and inserted into the database; other artifacts are copied into installation directory (e.g., Compliance Template).
User Data	C:\Documents and Settings\<username>\Local Settings\Application Data\HP\HP WebInspect	All data created by the user and not global for the application. Subdirectories include ComplianceTemplates, Exports, Logs, Plugins, Reporting, ScanData, and Tools.

3 Using WebInspect

Introduction

When you first start WebInspect, the application displays the **Start Page** tab in the client area, as illustrated below. This tab displays hypertext links to five major functions:

- Start a Web Site Scan
- Start a Web Service Scan
- Start an Enterprise Scan
- Generate a Report
- Start Smart Update



You can close the left pane by clicking the Left Arrow on the bar above the pane.

The contents of the right pane are determined by the button selected on the Button bar.

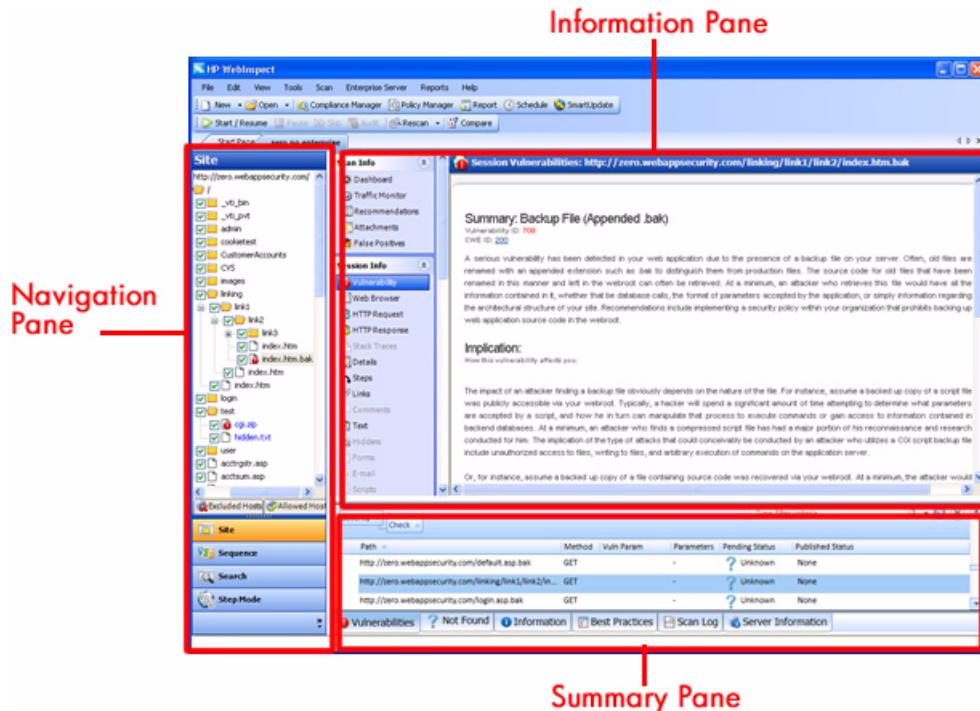


The choices are:

- **Home**—Displays a list of recently opened scans, as well as scans scheduled to be conducted today, recently generated reports, and messages downloaded from the HP server.
- **Manage Scans**—Displays a list of previously conducted scans, which you can open, rename, or delete. Click **Connections** to choose a database: either Local (scans stored in the SQL Server Express Edition database on your machine) or Remote (scans stored in a SQL Server Standard Edition database configured on your machine or elsewhere on the network), or both. See [Managing Scans](#) on page 113 for more information.
- **Manage Schedule**—Displays a list of scans that are scheduled to be performed. You can add a scan to the schedule, edit or delete a scheduled scan, or start the scan manually.

Each time you open or conduct a scan, WebInspect opens a tab labeled with the name or description of the target site. This work area is divided into three regions:

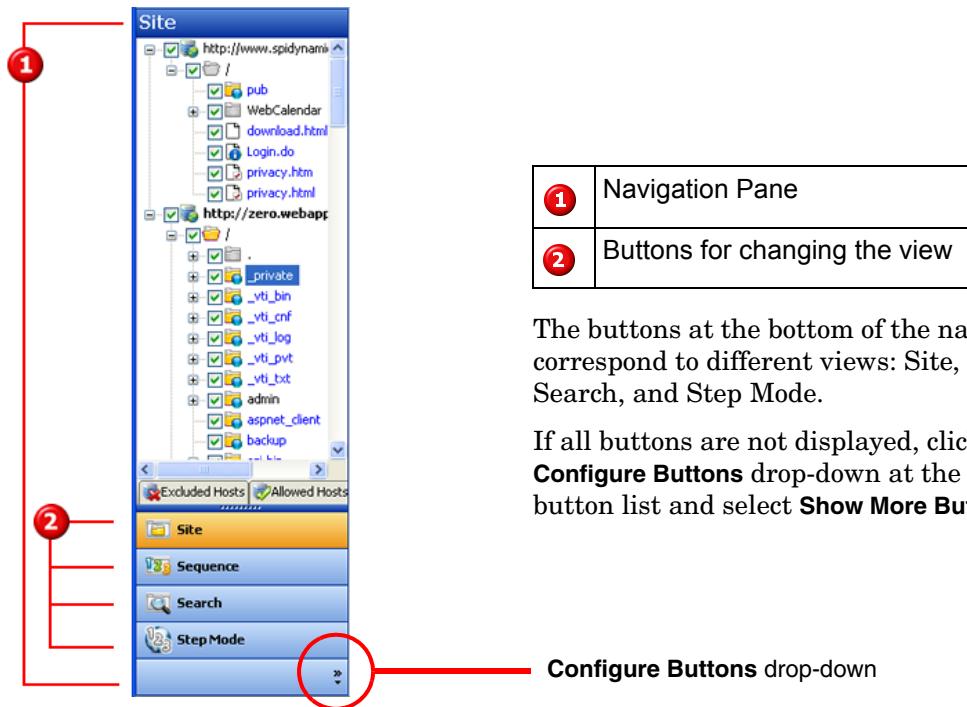
- Navigation Pane
- Information Pane
- Summary Pane.



If you have a large number of scans open at the same time, and there is no room to display all tabs, you can scroll the tabs by clicking the arrows on the extreme right end of the tab bar. Click the X to close the selected tab.

Navigation Pane

When conducting or viewing a scan, the navigation pane is on the left side of the *WebInspect* window. It includes the **Site**, **Sequence**, **Search**, and **Step Mode** buttons, which determine the contents (or “view”).



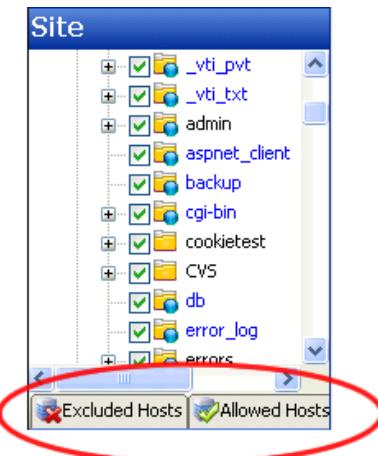
The buttons at the bottom of the navigation pane correspond to different views: Site, Sequence, Search, and Step Mode.

If all buttons are not displayed, click the **Configure Buttons** drop-down at the bottom of the button list and select **Show More Buttons**.

Site View

WebInspect displays in the navigation pane only those sessions that reveal the hierarchical structure of the Web site, plus those sessions in which a vulnerability was discovered. During the crawl of the site, WebInspect selects the check box next to each session (by default) to indicate that the session will also be audited. When conducting a sequential crawl and audit (where the site is completely crawled before being audited), you can exclude a session from the audit by clearing its associated check box before the audit begins.

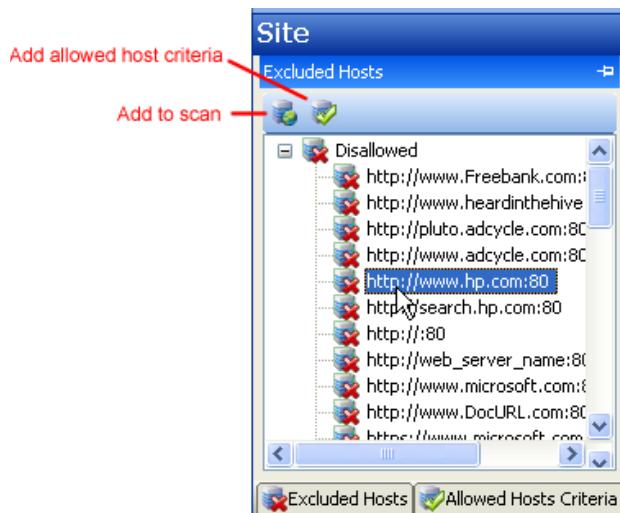
Site view also contains two pop-up tabs: **Excluded Hosts** and **Allowed Hosts Criteria**.



Excluded Hosts

If you click the **Excluded Hosts** tab (or hover your pointer over it), the tab displays a list of all disallowed hosts. These are hosts that may be referenced anywhere within the target site, but cannot be scanned because they are not specified in the Allowed Hosts setting (**Default/Current Scan Settings** → **Scan Settings** → **Allowed Hosts**).

Using the **Excluded Hosts** tab, you can select an excluded host and click either **Add to scan** or **Add allowed host criteria**.

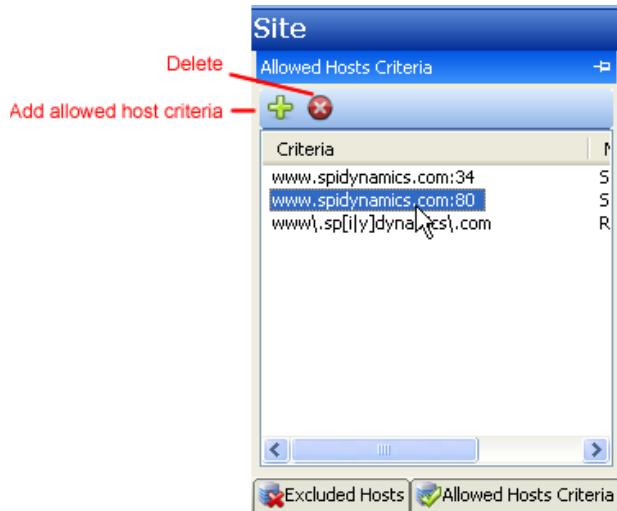


Adding a host to the scan creates a node in the site tree representing the host root directory. WebInspect will scan that session. If you have selected the option to log rejected sessions for invalid hosts (**Default/Current Scan Settings** → **Scan Settings** → **Session Storage**), WebInspect will scan the entire host.

Adding a host to the allowed host criteria adds the URL to the list of allowed hosts in the Current Scan Settings. WebInspect will include in the scan any subsequent links to that host. However, if you add a host to the allowed host criteria after WebInspect has already scanned the only resource containing a link to that host, the added host will not be scanned.

Allowed Hosts Criteria

If you click the **Allowed Hosts Criteria** tab (or hover your pointer over it), the tab displays the URLs (or regular expressions) specified in the WebInspect scan settings (under Allowed Hosts).



If you click either **Delete** or **Add allowed host criteria**, WebInspect opens the *Current Settings* window, where you can add, edit, or delete allowed host criteria (a literal URL or a regular expression representing a URL).

If you add an entry, WebInspect will include in the scan any subsequent links it encounters to hosts that match the criteria. However, if you specify a host after WebInspect has already scanned the only resource containing a link to that host, the added host will not be scanned. Similarly, if you delete an entry from the allowed host list, the scan will still include any resources that WebInspect already encountered.

To save these settings for a future scan, select **Save settings as** (at the bottom of the left pane of the *Settings* window).

You must pause the scan before you can modify the excluded hosts or allowed hosts criteria. Furthermore, the scanning of added or deleted hosts may not occur as expected, depending on the point at which you paused the scan. For example, if you add an allowed host after WebInspect has already scanned the only resource containing a link to the added host, the added host will not be scanned.

Sequence View

Sequence view displays server resources in the order they were encountered by WebInspect during a scan.

- In both Site view and Sequence view, blue text denotes a directory or file that was “guessed” by WebInspect, rather than a resource that was discovered through a link. For example, WebInspect always submits the request “GET /backup/ HTTP/1.1” in an attempt to discover if the target Web site contains a directory named “backup.”

Search View

Search view allows you to search across all sessions for various HTTP message components. For example, if you select **Response Raw** from the drop-down list and specify “set-cookie” as the search string, WebInspect lists every session whose raw HTTP response includes the “set-cookie” command.



To use the Search view:

- 1 In the navigation pane, click **Search** (at the bottom of the pane).
- 2 From the top-most list, select an area to search. The choices are:
 - Status Code
 - Request Raw
 - Request Post Data
 - Request Post Data Value
 - Request Header Name
 - Request Query
 - Request Query Value
 - Request Cookie Name
 - Request File Name and Extension
 - Request File Extension
 - Response Raw
 - Response Header Name
 - Response Cookies
 - Response Cookie Value
 - URL
 - Request Method
 - Request Post Data Name
 - Request Headers
 - Request Header Value
 - Request Query Name
 - Request Cookies
 - Request Cookie Value
 - Request File Name
 - Request Path
 - Response Headers
 - Response Header Value
 - Response Cookie Name
- 3 In the combo box, type or select the string you want to locate.
- 4 If the string represents a regular expression, select the **Regular Expression** check box.

- 5 To find an entire string in the HTTP message that exactly matches the search string, select the **Match Whole String** check box. The exact match is not case-sensitive.

➤ This option is not available for certain search targets.
- 6 Click **Search**.

Step Mode

Use Step Mode to navigate manually through the site, beginning with a session you select from either the site view or the sequence view.

Follow the steps below to step through the site:

- 1 In the site or sequence view, select a session.
- 2 Click the **Step Mode** button. If the button is not visible, click the **Configure Buttons** drop-down and select **Show More Buttons**.
- 3 When Step Mode appears in the navigation pane, select either **Audit as you browse** or **Manual Audit** from the **Audit Mode** list. **Manual Audit** is recommended



- 4 Click **Record**
 - 5 Click **Browse**.
- A browser opens and displays the response associated with the selected session. Continue browsing to as many pages as you like.
- 6 When done, return to WebInspect and click **Finish**.
- The new sessions are added to the navigation pane.
- 7 If you selected **Manual Audit** in step 3, click . WebInspect will audit all unaudited sessions, including those you added (or replaced) through Step Mode.

Navigation Pane Icons

Use the following table to identify resources displayed in the Sequence and Site views.

Icons Used on the Navigation Pane

Icon	Definition
	Server/host: Represents the top level of your site's tree structure.
	Blue folder: A private folder on your Web server found by WebInspect. These folders are not linked from the site itself.

Icons Used on the Navigation Pane (cont'd)

Icon	Definition
	Yellow folder: A folder whose contents are available over your Web site.
	Grey folder: A folder indicating the discovery of an item via path truncation. Once the parent is found, the folder will display in either blue or yellow, depending on its properties
	File.
	Query or Post.
	Document Object Model (DOM) event.
Icons superimposed on a folder or file indicate a discovered vulnerability	
	A red exclamation point indicates the object contains a critical vulnerability. An attacker might have the ability to execute commands on the server or retrieve and modify private information.
	A red dot indicates the object contains a high vulnerability. Generally, the ability to view source code, files out of the Web root, and sensitive error messages.
	A gold dot indicates the object contains a medium vulnerability. These are generally non-HTML errors or issues that could be sensitive.
	A blue dot indicates the object contains a low vulnerability. These are generally interesting issues, or issues that could potentially become higher ones.
	An "i" in a blue circle indicates an informational item. These are interesting points in the site, or certain applications or Web servers.
	A red check mark indicates a "best practice" violation.

Each object represents a session, which is a matched set comprising (a) the HTTP request sent by WebInspect to test for vulnerabilities and (b) the HTTP response from the server.

Navigation Pane Shortcut Menu

Right-clicking an item in the navigation pane displays a shortcut menu with the commands described in the following table.

Navigation Pane Shortcut Commands

Command	Definition
Expand Children *	(Site View only) Expands branching nodes in the site tree.
Collapse Children *	(Site View only) Contracts branching nodes into the superior node.

Navigation Pane Shortcut Commands (cont'd)

Command	Definition
Check All *	(Site View only) Marks the check box of all children.
Uncheck All *	(Site View only) Removes the check mark from all children.
Generate Session Report *	(Site View only) Creates a report showing summary information, the attack request and attack response, links to and from the URL, comments, forms, e-mail addresses, and check descriptions for the selected session.
Export Site Tree *	(Site View only) Saves the site tree in XML format to a location you specify.
Copy URL	Copies the URL of the selected session to the clipboard (the same as selecting Edit → Copy URL).
View in Browser	Displays the server's response in a Web browser.
Links	(Site View only) Lists all resources at the target site that contain links to the selected resource. The links may be rendered by HTML tags, scripts, or HTML forms. It also lists (under Linked To) all resources that are referenced by links within the HTTP response for the selected session. If you double-click a listed link, WebInspect shifts focus in the navigation pane to the referenced session. Alternatively, you can browse to the linked resource by viewing the session in the Web browser (click Web Browser).
Add	Allows you to add locations discovered by means other than a WebInspect scan (manual inspection, other tools, etc.) for information purposes. You can then add vulnerabilities to those locations so that a more complete picture of the site is archived for analysis. <ul style="list-style-type: none"> • Page - A distinct URL (resource). • Directory - A folder containing a collection of pages. <p>Choosing either Page or Directory invokes a dialog that allows you to name the directory or page and edit the HTTP request and response.</p> <ul style="list-style-type: none"> • Variation - A subnode of a location that lists particular attributes for that location. For example, the login.asp location might have the variation: "(Query) Username=12345&Password=12345&Action=Login" Variations are like any other location in that they can have vulnerabilities attached to them, as well as subnodes. Choosing Variation invokes the <i>Add Variation</i> dialog, allowing you to edit the variation attributes, specify Post or Query, and edit the HTTP request and response. • Vulnerability - A specific security threat. Choosing Vulnerability invokes the <i>Edit Vulnerabilities</i> dialog, allowing you to edit the variation attributes, specify Post or Query, and edit the HTTP request and response.

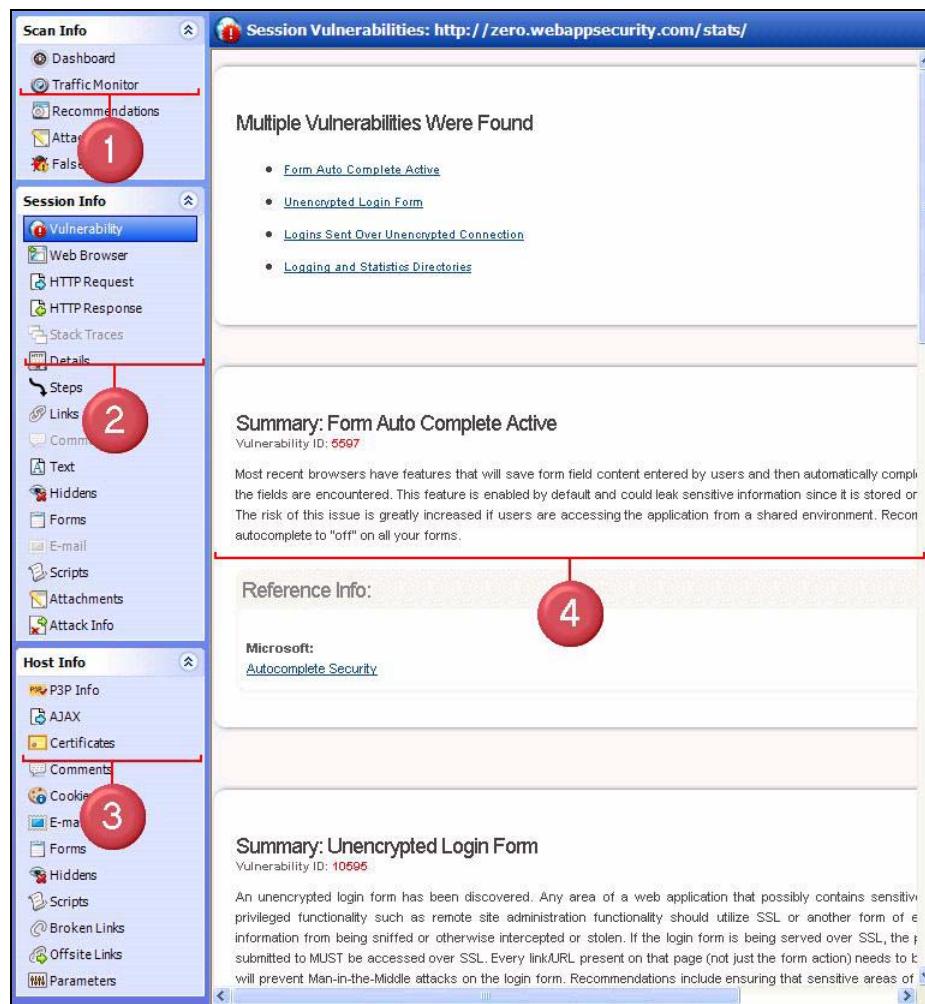
Navigation Pane Shortcut Commands (cont'd)

Command	Definition
Edit Vulnerabilities	Allows you to edit a location that was added manually or edit a vulnerability.
Remove Location	Removes the selected session from the navigation pane (both Site and Sequence views) and also removes any associated vulnerabilities. Note: You can recover removed locations (sessions) and their associated vulnerabilities. See Recovering Deleted Items on page 85 for details.
Review Vulnerability	Allows you to retest the vulnerability, mark it “ignored” or “false positive,” or send it to HP Quality Center or IBM Rational ClearQuest. For more information, see Retesting/Reviewing Vulnerabilities on page 85.
Mark as False Positive	Flags the vulnerability as a false positive and allows you to add a note.
Send to	Allows you convert the selected vulnerability to a defect and assign it to either HP Quality Center or IBM Rational ClearQuest, using the profile specified in the WebInspect application settings.
Remove Server	Deletes the server from the navigation pane and does not include the server in any remaining scan activity. This command appears only when you right-click a server.
Crawl	Recrawls the selected URL.
Attachments	Allows you to create a note associated with the selected session, flag the session for follow-up, add a vulnerability note, or add a vulnerability snapshot.
Tools	Presents a submenu of available tools.
Filter by Current Session	Restricts the display of items in the Summary pane to those having the SummaryDataID of the selected session.

* Command appears on shortcut menu only when the Navigation pane is using the Site view.

Information Pane

When conducting or viewing a scan, the information pane contains three collapsible information panels and an information display area.



① Scan Info Panel

② Session Info Panel

③ Host Info Panel

④ Information Display Area

Select the type of information to display by clicking on an item in one of the three information panels in the left column.



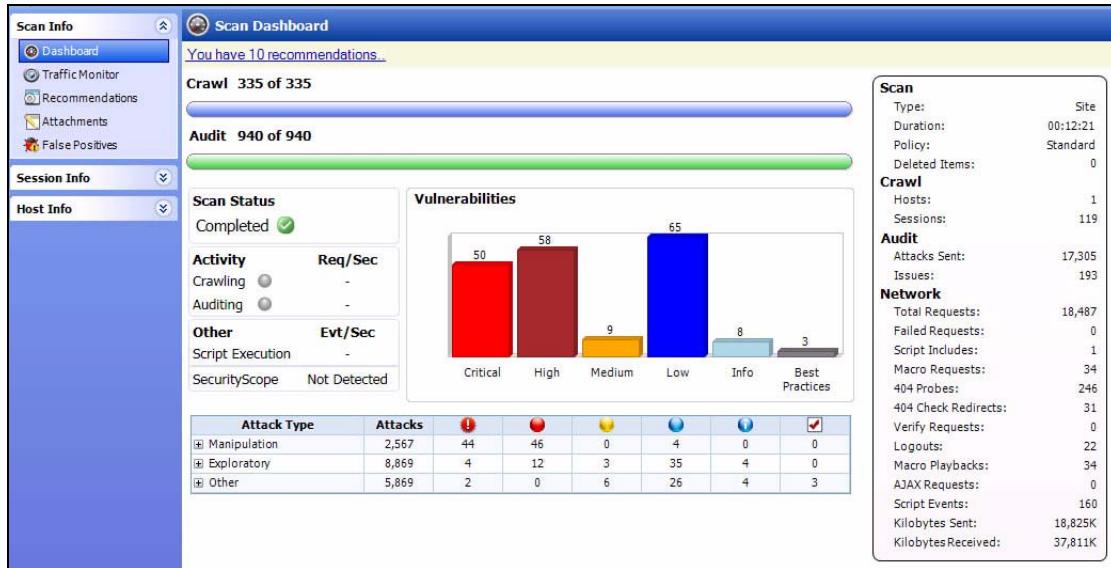
Tip: If you follow a link when viewing the vulnerability information, click the highlighted session in the navigation pane to return.

Scan Info Panel

The **Scan Info** panel has four default choices: **Dashboard**, **Recommendations**, **Attachments**, and **False Positives**. A fifth choice, **Traffic Monitor**, is displayed if you select this option from the Default settings.

Dashboard

The **Dashboard** selection displays a real-time summary of the scan results and a graphic representation of the scan progress.



Explanation of Dashboard Graphics

The following table describes the graphics used in the dashboard.

Dashboard Graphics

Graphic	Explanation
Crawl Gauge	Number of sessions crawled / Total Number of sessions for crawl.
Audit Gauge	Number of sessions audited / Total Number of sessions for audit.
Scan Status	Status: Running, Paused, or Complete.
Activity	<ul style="list-style-type: none"> Crawling - Number of requests per second the crawler is currently making (running average over last 5 seconds). Auditing - Number of requests per second the auditor is currently making (running average over last 5 seconds).
Other	<ul style="list-style-type: none"> Number of script events being executed per second (running average over last 5 seconds). HP SecurityScope indicator; reveals whether or not WebInspect has detected HP SecurityScope running on the target server.
Vulnerabilities Graph	Total number of issues identified for the scan sorted by severity level.
Attack Stats Grid	Number of attacks made and issues found, categorized by attack type and audit engine.

Explanation of Dashboard Statistics

The following table describes the statistics presented in the dashboard.

Dashboard Statistics

Group	Statistic	Explanation
Scan	Type	Type of scan: Site, Service, or Site Retest.
	Duration	Length of time scan has been running (can be incorrect if the scan terminates abnormally).
	Policy	Name of the policy used for the scan. For a retest, the field contains a dash (-), because the retest does not use the entire policy; see Retest All Vulnerabilities on page 156.
	Deleted Items	<p>Number of sessions and vulnerabilities removed by the user from the scan.</p> <p>To restore sessions or vulnerabilities that have been deleted:</p> <ol style="list-style-type: none"> 1 On the Scan Dashboard, click the number associated with deleted items. The <i>Recover Deleted Items</i> window appears. 2 Select either Vulnerabilities or Sessions from the drop-down list. 3 Select one or more items. 4 Click Recover.
Crawl	Hosts	Number of hosts included in the scan.
	Sessions	Total number of sessions (excluding AJAX requests, script and script frame includes, WSDL includes).
Audit	Attacks Sent	Total number of attacks sent.
	Issues	Total number of issues found (all vulnerabilities, as well as best practices).
Network	Total Requests	Total number of requests made.
	Failed Requests	Total number of failed requests.
	Script Includes	Total number of script includes.
	Macro Requests	Total number of requests made as part of macro execution.
	404 Probes	Number of probes made to determine file-not-found status.
	404 Check Redirects	Number of times a 404 probe resulted in a redirect.
	Verify Requests	Requests made for detection of stored parameters.
	Logouts	Number of times logout was detected and login macro executed.

Dashboard Statistics (cont'd)

Group	Statistic	Explanation
	Macro Playbacks	Number of times macros have been executed.
	AJAX Requests	Total number of AJAX requests made.
	Script Events	Total number of script events processed.
	Kilobytes Sent	Total number of kilobytes sent.
	Kilobytes Received	Total number of kilobytes received.

Traffic Monitor

WebInspect normally displays in the navigation pane only those sessions that reveal the hierarchical structure of the Web site, plus those sessions in which a vulnerability was discovered. The **Traffic Monitor** selection allows you to display and review every HTTP request sent by WebInspect and the associated HTTP response received from the server. To display this option, you must select **Enable Traffic Monitor Logging** in the Default settings (click **Edit** → **Default Settings**; then, in the Scan Settings category, select **General**).

Time	Host	Method	Url	Response Code	Engine
11/5/2010 1:45:45 PM	zero.webappsecurity....	GET	/pformresults.aspx?txtFirstName=%20<--XSS/*-*"/STYLE=xss;e/**>/expression(alert(...	200 OK	Query Inj...
11/5/2010 1:45:45 PM	zero.webappsecurity....	GET	/pformresults.aspx?txtFirstName=12345&txtLastName=><XSS/*-*"/STYLE=xss;e/**>...	200 OK	Query Inj...
11/5/2010 1:45:45 PM	zero.webappsecurity....	GET	/pformresults.aspx?txtFirstName=12345&txtLastName=%20<--XSS/*-*"/STYLE=xss;...	200 OK	Query Inj...
11/5/2010 1:45:45 PM	zero.webappsecurity....	GET	/pformresults.aspx?txtFirstName=12345&txtLastName=12345&txtHidden=%20<--X...	200 OK	Query Inj...
11/5/2010 1:45:45 PM	zero.webappsecurity....	GET	/pformresults.aspx?txtFirstName=12345&txtLastName=12345&txtHidden=This%20w...	200 OK	Query Inj...
11/5/2010 1:45:46 PM	zero.webappsecurity....	GET	/linking/baddir123/	404 Not Found	IIS Basic ...
11/5/2010 1:45:46 PM	zero.webappsecurity....	GET	/linking/baddir123/	404 Not Found	IIS No Ho...
11/5/2010 1:45:46 PM	zero.webappsecurity....	GET	/WebResource.axd?d=m9znum37VPoz2VM1z54ew2	404 Not Found	Request ...
11/5/2010 1:45:46 PM	zero.webappsecurity....	PROP...	/linking/baddir123/	501 Not Imple...	Request ...
11/5/2010 1:45:46 PM	zero.webappsecurity....	GET	/linking/baddir123/web.%c4%89onfig	404 Not Found	Request ...
11/5/2010 1:45:46 PM	zero.webappsecurity....	GET	/linking/baddir123/global.%C4%80sa	404 Not Found	Request ...

Buttons at the top of the information display area are described in the following table.

Traffic Monitor Buttons

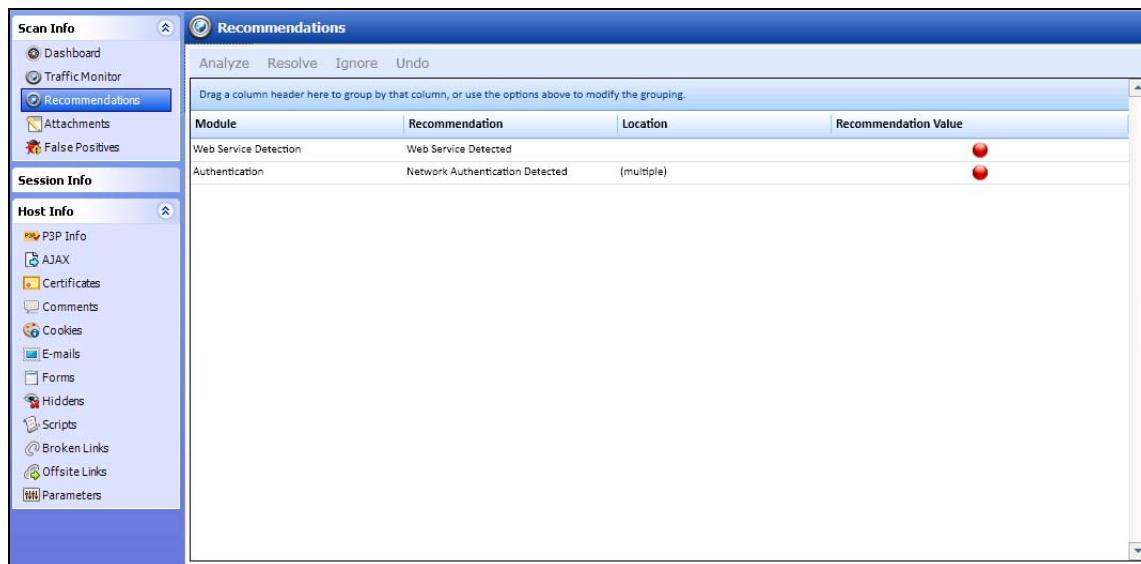
Button	Description
HTTP Editor	Opens the HTTP editor loaded with the selected request/response session.
Session	Offers three choices: <ul style="list-style-type: none"> • Navigate to Session: Navigate to the correlated session on this request in the site tree. • Navigate to Parent Session: Navigate to the correlated parent session on this request in the site tree. • Highlight Parent Session: Moves focus to the parent session of the selected session.

Traffic Monitor Buttons (cont'd)

Button	Description
Refresh	Updates display with most current information.
Auto Scroll	Automatically updates traffic monitor view with the latest traffic from WebInspect crawl and audit while scan is running. While in auto scroll mode, sorting is ascending by time, so user cannot sort without pausing the scan.
Columns	Allows user to select which traffic monitor database columns are displayed.

Recommendations

While conducting a scan, WebInspect may encounter certain omissions, abnormalities, or anomalies that interfere with or diminish the thoroughness of the assessment. WebInspect records this information and presents a list of recommendations designed to improve the quality of your scan when you next conduct it.



To enable this option, click the **Edit** menu and select **Default Scan Settings**; then, in the Scan Settings category, click **Recommendations** and select **Run Recommendation Modules when the scan is paused or completed**. Enabling this option while a scan is in progress will not activate the feature; it must be selected prior to beginning the scan.

This panel contains a list of recommendations, sorted by modules. Initially, all recommendations appear on the **Unresolved** tab. As you process each recommendation, the item is moved to either the **Resolved** or **Ignored** tab.

- 1 Double-click an item in the list of recommendations, or select an item and click **Resolve**. The *Recommendations* window appears.
- 2 If this window contains general information about recommendation modules, click **Next**.
- 3 The contents of the *Recommendations* window depend on the type of recommendation you selected (see details below).

- To discard the recommendation, click **Ignore**.
- To accept the recommendation, provide the requested information or perform the associated action (if required) and click **Resolve**.

Where applicable, modifications are saved to the current settings when you click **Resolve**. If you subsequently click **Undo** (to return the recommendation from Resolved to Unresolved), the settings are also restored to the original values.

- 4 Click **Next** to advance to the next recommendation.
- 5 Repeat steps 3-4 until all recommendations are addressed.
- 6 Click **Close**.
- 7 (Optional) If you resolved an issue, click **Scan** (in the Scan Recommendations information display area) to launch the Scan Wizard and rescan the site. Settings or actions resulting from resolved recommendations are incorporated into the scan settings.

Note: A scan created using a version of WebInspect prior to 9.0, when opened in the current version of WebInspect (9.30) will not contain recommendations.

[Authentication](#)

This module detects that network, proxy, or site authentication is required, but credentials are missing or invalid. See [Authentication](#) on page 192 for more information.

Action: Enter a valid user name and password.

[Web Macro](#)

This module determines that the macro may not be functioning properly. Usually this occurs because the macro is unable to log in or contains a poor log-out condition, or the site prevents multiple concurrent log-in sessions. For more information, see [Web Macro Recorder \(Traffic-Mode\)](#) on page 319.

Action: Click the “Authentication” hyperlink to access the scan settings for editing a login macro.

[File Not Found](#)

This module examines the server’s responses to requests for files and determines if the scan settings for recognizing a “file not found” condition are be incorrect. It is used only during a crawl-and-audit scan. For more information, see [File-Not-Found Settings](#) on page 194.

Action: Click the “File Not Found” hyperlink to modify the scan settings for detecting a file-not-found condition.

[Web Service](#)

The module detects Web service communications encountered during the scan. It displays summary information, including the number of WSDLs exposed for that site, the number of operations found, and the number of times those operations were invoked (along with the associated service method). You should conduct a Web Service scan on this site, in addition to the Web Site scan.

Action: Click **Design** to invoke the Web Service Test Designer tool, which will allow you to provide inputs to each operation.

Action: Click **Export** and select a file location. The program creates a folder for each URL and creates within that folder an xml file that contains a SOAP envelope, service URL, and operation name. You can incorporate this data when using the Web Service Test Designer to prepare for conducting a Web Service scan.

For more information, see [Web Service Test Designer](#) on page 420.

Form Values

This module detects forms containing an input element for which you have not provided a value. This could occur due to the following conditions:

- You did not select the option **Auto fill web forms during crawl** (see [Scan Settings - Method, Auto-fill Web forms during crawl](#) on page 166).
- You selected the option, but the form values file you specified does not contain an input element name that matches the name of the detected element.
- You selected the option and the specified form values file contains the name of the detected input element, but no value was specified.

To conduct a thorough analysis of your site, you should provide appropriate data for each input element. When WebInspect subsequently submits the form to an agent for processing, the application typically allows access to another page or section of the application. For the scanner to navigate through all possible links in the application, it must be able to submit appropriate data for each form.

On the *Form Values Detection* dialog, the left pane lists the name, value, and type of each input control on the form. The right pane (in most cases) depicts the detected form, as rendered in a browser. This pane also floats; if you need to view a larger representation, you can undock and reposition the frame.

Action: Supply the requested information using the left pane (although data for some element types may also be entered in the browser view).

If you did not previously select the option **Auto fill web forms during crawl**, WebInspect sets this option (in the Current Scan settings) and saves your input.

If you did select the option and specified a file, WebInspect appends your input to the file under a heading that identifies the specific URL at which the form was found.

In both cases, the information you supply will be used during a subsequent scan of the site.



Caution: Using Form Values recommendations can cause an unintended large increase in scan data stored, as well as potential “out of memory” errors during large scans. This module is turned off by default. If it is turned on, data storage problems and out of memory incidents may occur.

Custom Parameters

Custom Parameters are used to accommodate sites that use URL rewriting techniques and/or Representation State Transfer (REST) web services technologies. In addition to applying these rules that you discretely define or import, WebInspect will attempt (during a scan) to identify custom parameters and create rules to accommodate them. If you accept the recommended rules, they will be saved in the Custom Parameters settings.

For information on editing a rule, see [Custom Parameters](#) on page 183.

Attachments

The **Attachments** selection displays a list of all session notes, vulnerability notes, flags for follow-up, and vulnerability screenshots that have been added to the scan. Each attachment is associated with a specific session. This form also lists scan notes (that is, notes that apply to the entire scan rather than to a specific session).

You can create a scan note, or you can edit or delete an existing attachment.

To view an attachment, select the attachment and click **View** (or simply double-click the attachment).

To create a scan note, click the **Add** menu (in the information display area).

To edit an attachment, select the attachment and click **Edit**. Note that screenshots cannot be edited.

These functions are also available by right-clicking an attachment and selecting an option from the shortcut menu. You may also select **Go to session**, which opens the Session Info - Attachments pane and highlights in the navigation pane the session associated with that attachment.

To create attachments in other area of the WebInspect user interface, you can either:

- Right-click a session in the navigation pane and select **Attachments** from the shortcut menu, or
- Right-click a URL on the **Vulnerabilities** tab of the summary pane and select **Attachments** from the shortcut menu.

WebInspect automatically adds a note to the session whenever you send a defect to HP Quality Center or IBM Rational ClearQuest.

The screenshot shows the WebInspect interface with the 'Scan Notes' pane open. The pane title is 'Scan Notes: http://advm-asc-fservi.spidynamics.com:80'. It contains a table with two rows of data:

Type	Associated With	Date Added	Name	Size	URL
Note	Session	11/5/2010 2:25 PM	----	----	http://advm-asc-fservi.spidynamics.com:80
Screenshot	Vulnerability - Flash Cross-Domain Policy File	11/5/2010 2:28 PM	Screengrab	512 KB	http://advm-asc-fservi.spidynamics.com:80

Below the table, there is a 'Comments' section with a text input field containing 'Show this to project leader' and a 'Preview' section showing a small screenshot of a browser window.

False Positives

This feature lists all URLs that WebInspect originally flagged as containing a vulnerability and which a user later determined were false positives.

You can also import from a previous scan a list of vulnerabilities that were analyzed as being false positive. WebInspect then correlates these false positives from a previous scan with vulnerabilities detected in the current scan and flags the new occurrences as false positives.

To illustrate, suppose a cross-site scripting vulnerability was detected in Scan No. 1 at URL <http://www.mysite.com/foo/bar> and, after further analysis, someone flagged it as a false positive. If you import false positives from Scan No. 1 into Scan No. 2 of www.mysite.com, and if that second scan detects a cross-site scripting vulnerability at the same URL (<http://www.mysite.com/foo/bar>), then WebInspect automatically changes that vulnerability to a false positive.

Imported false positives are loaded first into a list labeled “Inactive False Positives.” If a false positive in that list is matched with a vulnerability in the current scan, the item is moved from the Inactive False Positives list to the Active False Positives list. Unmatched items remain in the Inactive False Positives list.

False positives from other scans can be manually loaded into the current scan at any time. Alternatively, you may instruct the Scan Wizard, while initiating a scan, that false positives are to be loaded from a specific file; in this case, WebInspect correlates the false positives as they are encountered during the scan. You can also see (on the scan dashboard) the false positives matched while the scan is running.

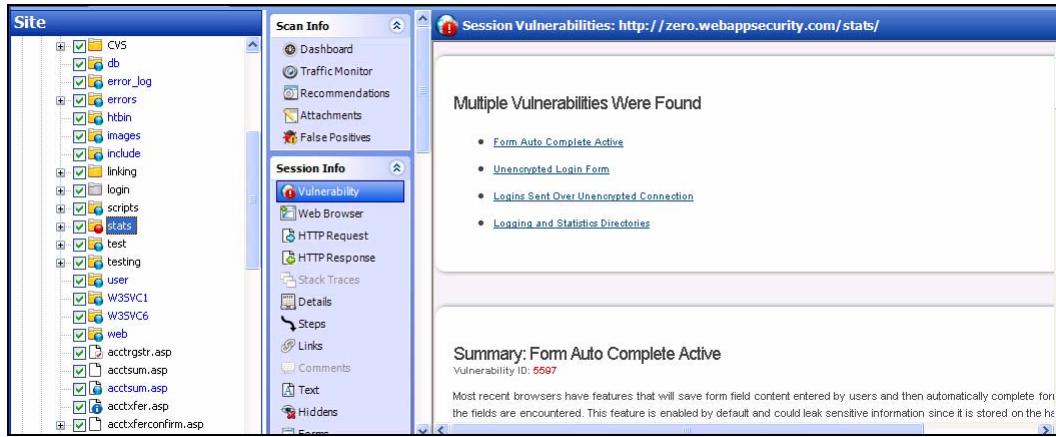
Risk	Source	URL	Vulnerability	State
	Zero (Current Scan)	http://zero.webappsec...	Microsoft Active Server Pages Cookie Retrieval Issue (1 item)	Active False Positives
	Zero (Current Scan)	http://zero.webappsec...	IIS Global Server Variables Disclosure (global.asa.bak) (1 item)	Active False Positives

- 1 Select **False Positives** from the Scan Info panel.
- 2 If necessary, click the plus sign next to a vulnerability description to display the associated URLs and state.
- 3 Click a URL to view a comment (at the bottom of the Information pane) that may have been entered when the user removed the vulnerability.
- 4 To import false positives from other scans, click **Import Scans**.
- 5 To change a false positive back to a vulnerability, select an item from the Active False Positive list and click **Mark as Vulnerability**.
- 6 To remove an item from the Inactive False Positive list, select the item and click **Remove From Inactive**.
- 7 To edit a comment associated with a false positive, select the item and click **Edit Comment**.

For information on how to designate a vulnerability as a false positive, see [Navigation Pane Shortcut Menu](#) on page 64 and [Vulnerabilities Tab](#) on page 82.

Session Info Panel

WebInspect lists each session created during a scan in the navigation pane using either the Site view or Sequence view. Select a session and then click one of the options in the **Session Info** panel to display related information about that session.



In the above example scan, WebInspect sent the HTTP request GET /stats/stats.html HTTP/1.1. To see the vulnerability:

- 1 Select **Stats.html** in the navigation pane.
- 2 Click **Vulnerability** in the **Session Info** panel.

The following table lists the options available in the **Session Info** panel. Some options appear only for specific scans (Web Site Scan or Web Service Scan). Also, options are enabled only if they are relevant to the selected session; for example, the **Forms** selection is not available if the session does not contain a form.

Options in Session Info Panel

Option	Definition
Vulnerability	Displays the vulnerability information for the session selected in the navigation pane.
Web Browser ¹	Displays the server's response as rendered by a Web browser for the session selected in the navigation pane.
HTTP Request	Displays the raw HTTP request sent by WebInspect to the server hosting the site you are scanning.
HTTP Response	Displays the server's raw HTTP response to WebInspect's request. If the response contains one or more attack signatures (indicating that a vulnerability has been discovered) you can tab from one attack signature to the next by clicking these buttons:
	Note: If you select a Flash (.swf) file, WebInspect displays HTML instead of binary data. This allows WebInspect to display links in a readable format.

Options in Session Info Panel (cont'd)

Option	Definition
Stack Traces	<p>This feature is designed to support HP Fortify SecurityScope when it is installed and running on the target server.</p> <p>For certain checks (such as SQL injection, command execution, and cross-site scripting), SecurityScope intercepts WebInspect HTTP requests and conducts runtime analysis on the target module. If this analysis confirms that a vulnerability exists, SecurityScope appends the stack trace to the HTTP response. Developers can analyze this stack trace to investigate areas that require remediation.</p>
Details ¹	<p>Lists request and response details, such as the size of the response and the request method. Note that the Response section contains two entries for content type: returned and detected. The Returned Content Type indicates the media type specified in the Content-Type entity-header field of the HTTP response. Detected Content Type indicates the actual content-type as determined by WebInspect.</p>
Steps ¹	<p>Displays the route taken by WebInspect to arrive at the session selected in the navigation pane or the URL selected in the summary pane. Beginning with the parent session (at the top of the list), the sequence reveals the subsequent URLs visited and provides details about the scan methodology.</p>
Links ¹	<p>This option lists (under Linked From) all resources at the target site that contain links to the selected resource. The links may be rendered by HTML tags, scripts, or HTML forms. It also lists (under Linked To) all resources referenced by links within the HTTP response for the selected session.</p>
Comments ¹	<p>Displays all comments (in HTML) embedded in the HTTP response.</p>
Text ¹	<p>Displays all text contained in the HTTP response.</p>
Hiddens ¹	<p>Displays the name of each input element whose control type is "hidden."</p>
Forms ¹	<p>Displays the HTML interpreted by the browser to render forms.</p>
E-mail ¹	<p>Displays all e-mail addresses included in the response.</p>
Scripts ¹	<p>Displays all client-side scripts embedded in the server's response.</p>
Attachments	<p>Displays all notes, flags, and screenshots associated with the selected object.</p> <p>To create an attachment, you can either:</p> <ul style="list-style-type: none"> • Right-click a session (Web site scan) or an operation or vulnerability (Web service scan) in the navigation pane and select Attachments from the shortcut menu, or • Right-click a URL on the Vulnerabilities tab of the summary pane and select Attachments from the shortcut menu, or • Select a session (Web site scan) or an operation or vulnerability (Web service scan) in the navigation pane, then select Attachments from the Session Info panel and click the Add menu (in the information pane). <p>WebInspect automatically adds a note to the session information whenever you send a defect to HP Quality Center or IBM Rational ClearQuest.</p>

Options in Session Info Panel (cont'd)

Option	Definition
Attack Info ¹	Displays the attack sequence number, URL, name of the audit engine used, and the result of the vulnerability test. Attack information is usually associated with the session in which the attack was created and not with the session in which it was detected. If attack information does not appear for a selected vulnerable session, select the parent session and then click Attack Info . Also, attack information for non-vulnerable sessions will not appear unless you have enabled the appropriate session storage option in the default settings; see Session Storage on page 176 for more information.
XML Request ²	Displays the associated XML schema embedded in the request (available when selecting the WSDL object during a Web Service Scan).
XML Response ²	Displays the associated XML schema embedded in the response (available when selecting the WSDL object during a Web Service Scan).
Web Service Request ²	Displays the web service schema and values embedded in the request (available when selecting an operation during a Web Service Scan).
Web Service Response ²	Displays the web service schema and values embedded in the response (available when selecting an operation during a Web Service Scan).

¹ Web Site Scan only

² Web Service Scan only

Most options provide a Search feature at the top of the information pane, allowing you to locate the text you specify. To conduct a search using regular expressions, select the **Regex** option before clicking **Find**.



Tip: If you follow a link when viewing the vulnerability information, click the highlighted session in the navigation pane to return.

Host Info Panel

When you click any item listed in this collapsible panel, WebInspect displays all instances of that item type that were discovered during a crawl or audit of the site (or host). For example, in the following illustration, the user selected **Cookies** in the **Host Info** panel and WebInspect listed all sessions in which cookies were used.

The screenshot shows the WebInspect interface with the 'Host Info' panel expanded. Under the 'Cookies' category, there are several items listed: Start Macro, HTML, HTML, HTML, Attack Redirect, Path Truncation, HTML, HTML, and HTML. Each item has a URL and Post Data associated with it. Below the list, there is a snippet of raw cookie data:

```
Set-Cookie: status=yes; path=/  
Set-Cookie: username=admin; path=/  
Set-Cookie: userid=admin; path=/  
Set-Cookie: sessionid=C9C006CB3F9B8B111F4D2C8AB5EF7AA50001; path=/  
Set-Cookie: ASPSESSIONIDCCTQRBSD=KLACDIOCFHNGIKDIDKDGMEAE; path=/
```

If you double-click an item in the list, WebInspect highlights in the navigation pane the session that contains that item. You can copy items (such as e-mail addresses) to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

► Note: The Host Info panel is not displayed when conducting a Web Service scan.

In most cases, you can use the Search feature at the top of the information pane to locate the text you specify. To conduct a search using regular expressions, select the **Regex** button before clicking **Find**.

Options in Host Info Panel

Option	Description
P3P Info	The World Wide Web Consortium's Platform for Privacy Preferences Project (P3P) enables Web sites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents. P3P user agents allow users to be informed of site practices (in both machine- and human-readable formats) and to automate decision-making based on these practices when appropriate. Thus users need not read the privacy policies at every site they visit. A P3P-compliant Web site declares in a policy the kind of information it collects and how that information will be used. A P3P-enabled Web browser can decide what to do by comparing this policy with the user's stored preferences. For example, a user may set browser preferences so that information about their browsing habits should not be collected. When the user subsequently visits a Web site whose policy states that a cookie is used for this purpose, the browser automatically rejects the cookie.

Options in Host Info Panel (cont'd)

Option	Description
AJAX	<p>AJAX is an acronym for Asynchronous JavaScript and XMLHttpRequest. If you select this option, WebInspect displays all pages containing an AJAX engine, as well as the AJAX requests.</p> <p>AJAX is not a technology per se, but a combination of existing technologies, including HTML or XHTML, Cascading Style Sheets, JavaScript, the Document Object Model, XML, XSLT, and the XMLHttpRequest object. When these technologies are combined in the AJAX model, Web applications are able to make quick, incremental updates to the user interface without reloading the entire browser page.</p> <p>Instead of loading a Web page at the start of the session, the browser loads an AJAX engine that is responsible for both rendering the user interface and communicating with the server. Every user action that normally would generate an HTTP request takes the form of a JavaScript call to the AJAX engine instead. Any response to a user action that does not require communication with the server (such as simple data validation, editing data in memory, and even some navigation) is handled by the engine. If the engine needs to communicate with the server — submitting data for processing, loading additional interface code, or retrieving new data — the engine makes those requests asynchronously, usually using XML.</p>
Certificates	<p>A certificate states that a specific Web site is secure and genuine. It ensures that no other Web site can assume the identity of the original secure site. When sending personal information over the Internet, users should check the certificate of the Web site to ensure that it will protect personally identifiable information. When downloading software from a Web site, certificates verify that the software is coming from a known, reliable source.</p> <p>A certificate associates an identity with a public key. Only the owner of the certificate knows the corresponding private key, which allows the owner to make a “digital signature” or decrypt information encrypted with the corresponding public key.</p>
Comments	Often, developers leave critical information in comments that can be used to breach the security of a site.
Cookies	A cookie contains information (such as user preferences or configuration information) stored by a server on a client for future use. Cookies appear in two basic forms: as individual files or as records within one contiguous file. Often, there are multiple sets, the result of multiple browsers being installed in differing locations. In many cases, “forgotten” cookies contain revealing information that you would prefer others not see.
E-mails	WebInspect lists all e-mail addresses discovered during a scan.
Forms	WebInspect lists all HTML forms discovered during a scan.
Hiddens	WebInspect analyzes all forms and then lists all controls of the type “hidden” (i.e., controls that are not rendered but whose values are submitted with a form). Developers often include parameters in hidden controls that can be edited and resubmitted by an attacker.

Options in Host Info Panel (cont'd)

Option	Description
Scripts	A script is a program that runs on a server and processes requests based on input from the browser. WebInspect tracks each script as it runs, and lists all scripts on the Information tab.
Broken Links	WebInspect finds and documents all non-working hyperlinks on the site.
Offsite Links	WebInspect finds and documents all hyperlinks to other sites.
Parameters	WebInspect finds and documents all parameters encountered during the scan. A parameter is either a query string submitted as part of the URL in the HTTP request (or contained in another header) or data submitted using the Post method.

Summary Pane

When conducting or viewing a scan, use the horizontal summary pane at the bottom of the window to see a centralized display of discovered vulnerabilities. It allows you to access vulnerability information quickly and view WebInspect logging information. This pane has six tabs:

- Vulnerabilities
- Not Found
- Information
- Best Practices
- Scan Log
- Server Information

The screenshot shows the WebInspect interface with the 'Summary' tab selected. At the top, there are two dropdown menus: 'Severity' and 'Check'. To the right is a search bar with placeholder text 'Type filter criteria...' and various icons. Below these are six tabs: 'Vulnerabilities' (selected), 'Not Found', 'Information', 'Best Practices', 'Scan Log', and 'Server Information'. The main content area displays a table of findings grouped by check type. Under 'Check:Backup File (cgi.zip) (3 items)', there are three entries for 'http://zero.webappsecurity.com/admin/cgi.zip' and 'http://zero.webappsecurity.com/cgi.zip'. Under 'Check:Cross-Site Scripting (36 items)', there are two entries for 'http://zero.webappsecurity.com/acctxferconfirm.asp'. The table columns include Path, Method, Vuln Param, Parameters, Pending Status, and Published Stat.

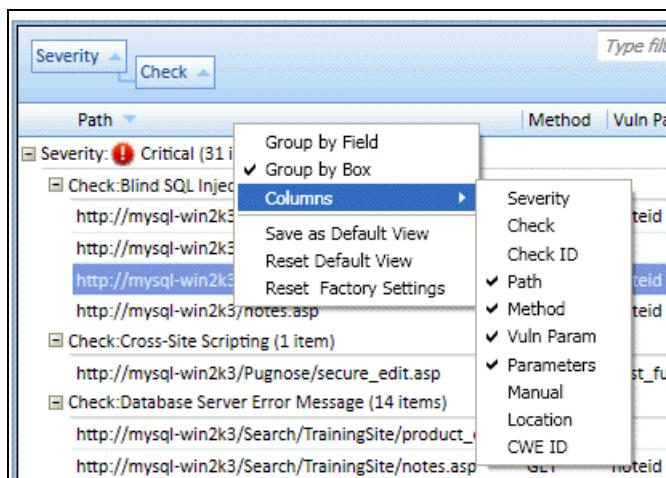
Path	Method	Vuln Param	Parameters	Pending Status	Published Stat	
http://zero.webappsecurity.com/admin/cgi.zip	GET	-	?	Unknown	None	
http://zero.webappsecurity.com/cgi.zip	GET	-	?	Unknown	None	
http://zero.webappsecurity.com/test/cgi.zip	GET	-	?	Unknown	None	
http://zero.webappsecurity.com/acctxferconfirm.asp	POST	fromAcct	(Post)from...	?	Unknown	None
http://zero.webappsecurity.com/acctxferconfirm.asp	POST	amount	(Post)from...	?	Unknown	None

Note: You can also group and filter results on all tabs except Scan Log. For more information, see [Using Filters and Groups in the Summary Pane](#) on page 88.

Vulnerabilities Tab

The **Vulnerabilities** tab lists information about each vulnerability that WebInspect during an audit of your Web presence.

To select the information you want to display, right-click the column header bar and choose **Columns** from the shortcut menu.



The available columns are:

- Severity: A relative assessment of the vulnerability, ranging from low to critical. See below for associated icons.
- Check: A WebInspect probe for a specific vulnerability, such as cross-site scripting, unencrypted log-in form, etc.
- Check ID: The identification number of a WebInspect probe that checks for the existence of a specific vulnerability. For example, Check ID 742 tests for database server error messages.
- Path: The hierarchical path to the resource.
- Method: The HTTP method used for the attack.
- Stack: Stack trace information obtained from HP Fortify Software Security Center. This column is available only when SecurityScope is enabled during the scan.
- Vuln Param: The name of the vulnerable parameter.
- Parameters: Names of parameters and values assigned to them.
- Published Status: The status as it exists in Software Security Center, if previously published.
- Pending Status: The status (assigned automatically by WebInspect or manually) if this scan were to be published.
- Manual: Displays a check mark if the vulnerability was manually created.
- Duplicates: Vulnerabilities detected by SecurityScope that are traceable to the same source.
- Location: Path and data that revealed the vulnerability.
- CWE ID: The Common Weakness Enumeration identifier(s) associated with the vulnerability.

- **Kingdom:** The category in which this vulnerability is classified, using a taxonomy of software security errors developed by the Fortify Software Security Research Group. Click the hyperlink to navigate to <http://www.hpenterprisesecurity.com/vulncat/en/vulncat/index.html>
- **Reproducible:** Values may be Reproduced, Not Found/Fixed, or New. This column is available for Site Retests only (Verify Vulnerabilities).

The severity of vulnerabilities is indicated by the following icons.

Critical	High	Medium	Low

If you click an item in the list, the program highlights the related session in the navigation pane and displays associated information in the information pane.

With a session selected, you can also view associated information by selecting an option from the Session Info panel.

For Post and Query parameters, click an entry in the Parameters column to display a more readable synopsis of the parameters.

Right-clicking an item in the list displays a shortcut menu with the commands described in the following table.

Vulnerability Shortcut Menu

Command	Definition
Copy URL	Copies the URL to the Windows clipboard.
Copy Selected Item(s)	Copies the text of selected items to the Windows clipboard.
Copy All Items	Copies the text of all items to the Windows clipboard
Export	Creates a comma-separated values (csv) file containing either all items or selected items and displays it in Microsoft Excel.
View in Browser	Renders the HTTP response in a browser.
Filter by Current Value	Restricts the display of vulnerabilities to those that satisfy the criteria you select. For example, if you right-click on “Post” in the Method column and then select Filter by Current Value , the list displays only those vulnerabilities that were discovered by sending an HTTP request that used the Post method. Note that the filter criterion is displayed in the combo box in the upper right corner of the summary pane. Alternatively, you can manually enter or select a filtering criterion using this combo box. For additional details and syntax rules, see Using Filters and Groups in the Summary Pane on page 88.
Modify Publish Status	Change the status of a vulnerability/issue before publishing to HP Fortify Software Security Center. This command is available only when connected to WebInspect Enterprise.
Change Severity	Allows you to change the severity level.

Vulnerability Shortcut Menu (cont'd)

Command	Definition
Edit Vulnerability	Allows you to add, delete, or edit a vulnerability associated with the selected session.
Review Vulnerability	Allows you to retest the vulnerable session, mark it as false positive or ignored, or send it to HP Quality Center or IBM Rational ClearQuest. For more information, see Retesting/Reviewing Vulnerabilities on page 85. This option is also invoked if you double-click a vulnerability.
Mark As	<p>Allows you to flag the vulnerability as either a false positive or ignored.</p> <p>To view false positives, select False Positives in the Scan Info panel.</p> <p>To view (and optionally recover) ignored vulnerabilities, select Dashboard in the Scan Info panel and click the number associated with deleted items. When the <i>Recover Deleted Items</i> window appears, select Vulnerabilities from the drop-down list.</p>
Send To	Converts the vulnerability to a defect and adds it to the HP Quality Center or IBM Rational ClearQuest database.
Remove Location	<p>Deletes the session from the navigation pane.</p> <p>Note: You can recover removed sessions. For information, see Recovering Deleted Items on page 85.</p>
Crawl	Recrawls the selected URL.
Tools	Presents a submenu of available tools.
Attachments	Allows you to create a note associated with the selected session, flag the session for follow-up, add a vulnerability note, or add a vulnerability snapshot.

If you right-click a group heading, a shortcut menu allows you to:

- Collapse/Expand All Groups
- Collapse/Expand Group
- Copy Selected Item(s)
- Copy All Items
- Change Severity
- Mark as
- Send to
- Remove Location

Recovering Deleted Items

When you remove a session or “ignore” a vulnerability, WebInspect deletes the item from the Navigation pane (in both the Site and Sequence views) and from the **Vulnerability** tab in the Summary pane. It also omits those items from any reports you may generate. The number of deleted items is displayed on the Dashboard (under the Scan category). You can recover removed sessions and ignored vulnerabilities using the following procedure.

- 1 Click the highlighted number that appears next to the Deleted Items header.
The *Recover Deleted Items* window displays a list of deleted sessions or deleted vulnerabilities.
- 2 Click the drop-down list to toggle between ignored vulnerabilities and removed sessions.
- 3 Select the check box next to one or more items you want to recover.
- 4 To view detailed information about the items, select **Show details when selected**.
- 5 Click **Recover** and then click **Yes** when prompted to verify your selection.

Recovered vulnerabilities reappear in the Navigation pane in both the Site and Sequence views (along with their parent sessions) and also reappear in the **Vulnerabilities** tab in the Summary pane. Recovered sessions also reappear in the Navigation pane along with any child sessions and their vulnerabilities.

Retesting/Reviewing Vulnerabilities

After you conduct a scan and report discovered vulnerabilities, developers may correct their code and update the site. You can then open the original scan, select the once-vulnerable session (now supposedly remediated), and select **Review Vulnerability** from the shortcut menu. Assuming that the fundamental architecture of the site has not changed, you can verify that the threat no longer exists without rescanning the entire site (which, in some cases, could require several hours or even days).

Alternatively, you can use this feature simply to double-check a reported vulnerability, even while the scan is still running.

- 1 Right-click a vulnerable session from the Navigation pane (or right-click a URL on the **Vulnerability** tab of the Summary pane).
- 2 Select **Review Vulnerability** from the shortcut menu.
The *Review Vulnerability* window opens.
- 3 If you want to access the site through Web Proxy, click **Options** and select **Launch and Direct Traffic through Web Proxy**.
- 4 If multiple vulnerabilities are associated with the selected session, choose one from the **Vulnerabilities to Test** list.
- 5 Use the tabs on the right side of the client area to display information about the original session (as selected in the lower pane under the **URL** column):
 - Browser - The server’s response, as rendered in a browser.
 - Request - The raw HTTP request message.
 - Response - The raw HTTP response message.
 - Vulnerability - A description of the vulnerability, its implications, and suggestions on how to fix it.
- 6 To retest the session for the selected vulnerability, click **Retest**.

Results of the retest appear on the Status bar and in the lower pane in the **Response Match Status** column.

The status is reported as either “Vulnerability <VulnerabilityName> Detected” or “Vulnerability <VulnerabilityName> Not Detected.”

The reliability of the reported findings is mitigated by the Response Match Status, which may have the following values:

- Match - The resource has not changed significantly; WebInspect was able to access the session via the same path used by the original scan.

- Inconclusive - Based on the HTTP response, the resource has changed in a manner that may or may not substantiate the finding that a vulnerability has or has not been detected during the retest.

- Different - The HTTP response is radically different from the response received during the original scan, suggesting major changes to the resource.

- 7 If you think that WebInspect has erroneously determined that the vulnerability exists, you can remove the vulnerability by clicking **Mark as False Positive**.
- 8 To ignore the vulnerability, click **Mark as** and select **Ignored** from the drop-down list.
- 9 To convert one or more vulnerabilities to defects and add them to either the HP Quality Center or IBM Rational ClearQuest database, click **Send To**.

Note: If you access the *Vulnerability Review* window from the *Vulnerability Compare* window, the **Mark As** and **Send To** buttons are not enabled.

Not Found Tab

This tab is germane only when WebInspect is connected to WebInspect Enterprise or to the Assessment Management Platform (AMP) for the purpose of synchronizing and uploading vulnerability data with HP Fortify Software Security Center. It lists vulnerabilities that were detected by a previous scan in a specific project version, but were not detected by the current scan. These vulnerabilities are not included in counts on the dashboard and are not represented in the site or sequence view of the navigation pane.

For more information, see [Publishing to Software Security Center](#) on page 105.

The shortcut menu options, grouping, and filtering capabilities are the same as described for the **Vulnerabilities** tab.

Information Tab

The **Information** tab lists information discovered during a WebInspect scan. These are not considered vulnerabilities, but simply identify interesting points in the site or certain applications or Web servers. When you click a listed URL, the program highlights the related item in the navigation pane.

The shortcut menu options, grouping, and filtering capabilities are the same as described for the **Vulnerabilities** tab.

Best Practices Tab

The **Best Practices** tab lists issues detected by WebInspect that relate to commonly accepted best practices for Web development. Items listed here are not vulnerabilities, but are indicators of overall site quality and site development security practices (or lack thereof).

The shortcut menu options, grouping, and filtering capabilities are the same as described for the **Vulnerabilities** tab.

Scan Log Tab

Use the **Scan Log** tab to view information about your WebInspect scan action. For instance, the time at which certain audit methodologies are applied against your Web presence are listed here. You can select the logging level (Debug, Info, Warn, Error, or Fatal) using the **Logging** option on the *Application Settings* window.

Time	Level	Message
6/27/2012 8:19:25 AM	Info	Scan Start, ScanID:941fb508-4d44-4ea9-a3d3-0861c7367b2f Version:9.30.43.0, Location:C:\...
6/27/2012 8:21:15 AM	Info	Scanner Retry Start:
6/27/2012 8:21:15 AM	Info	Scanner Retry Stop:
6/27/2012 8:21:16 AM	Info	Recommendation Modules Started:
6/27/2012 8:21:16 AM	Info	Starting Recommendation Module: Web Macro:
6/27/2012 8:21:16 AM	Info	Starting Recommendation Module: Network Authentication:
6/27/2012 8:21:16 AM	Info	Starting Recommendation Module: File Not Found:
6/27/2012 8:21:16 AM	Info	Completed Recommendation Module: File Not Found:
6/27/2012 8:21:16 AM	Info	Completed Recommendation Module: Network Authentication:
6/27/2012 8:21:16 AM	Info	Starting Recommendation Module: Web Service:

You can select the logging level (Debug, Info, Warn, Error, or Fatal) using the Logging option on the *Application Settings* window.

You can filter the type of messages displayed using the **Errors**, **Warnings**, and **Messages** buttons at the top of the pane. To view detailed information about a specific entry in the scan log, select an entry and then click **Detail**.

You can also right-click an entry and select the following options from the shortcut menu:

- Copy selected row clipboard
- Copy all items to clipboard
- Get more information about this message

Server Information Tab

This tab lists items of interest pertaining to the server. Only one occurrence of an item or event is listed.

Using Filters and Groups in the Summary Pane

Using Filters

You can display a subset of items that match the criteria you specify using either of two methods:

- Enter filter criteria using the combo box in the top right corner of the pane.
- Note: Click the filter criteria box and press CTRL + Space to view a pop-up list of all available filter criteria, and then enter a value for that criterion.
- Right-click a value in any column and select **Filter by Current Value** from the shortcut menu.

This filtering capability is available on all **Summary** pane tabs except **Scan Log**.

In the following example, the first screen shows unfiltered items on the **Vulnerabilities** tab. The second screen is rendered entering “Method:Get” in the filter criteria box.

No Filter

The screenshot shows the 'Vulnerabilities' tab of the Summary pane. The filter criteria box at the top right contains 'Severity'. The table below lists various vulnerabilities, including several cross-site scripting (XSS) attacks and a backup file vulnerability. The 'Path' column shows URLs like 'http://zero.webappsecurity.com/test/cgi.zip' and 'http://zero.webappsecurity.com/admin.cgi.zip'. The 'Method' column shows 'GET' or 'POST' for most entries. The 'Vuln Param' column contains parameters such as 'fromAcct', 'amount', 'toAcct', 'err', and '(Query)err...'. The 'Published Status' column indicates that all items are 'Unknown' and 'None'.

Path	Method	Vuln Param	Parameters	Pending Status	Published Status
http://zero.webappsecurity.com/test/cgi.zip	GET	-	?	Unknown	None
Check:Cross-Site Scripting (36 items)					
http://zero.webappsecurity.com/acctxferconfirm.asp	POST	fromAcct	(Post)from...	?	Unknown
http://zero.webappsecurity.com/acctxferconfirm.asp	POST	amount	(Post)from...	?	Unknown
http://zero.webappsecurity.com/acctxferconfirm.asp	POST	toAcct	(Post)from...	?	Unknown
http://zero.webappsecurity.com/banklogin.asp	GET	err	(Query)err...	?	Unknown
http://zero.webappsecurity.com/cookiestest>ShowCook...	GET	-	?	Unknown	None
http://zero.webappsecurity.com/cookiestest>ShowCook...	GET	-	?	Unknown	None
Severity	Check	Type filter criteria...			
Vulnerabilities	Not Found	Information	Best Practices	Scan Log	Server Information

Filtered by Method: Get

The screenshot shows the 'Vulnerabilities' tab of the Summary pane with a filter applied. The filter criteria box at the top right contains 'Method:Get'. The table below shows the same set of vulnerabilities as the previous screenshot, but only the ones with 'Method' set to 'GET' are displayed. This includes the backup file vulnerability and the XSS attacks mentioned earlier.

Path	Method	Vuln Param	Parameters	Pending Status	Published Status
http://zero.webappsecurity.com/admin.cgi.zip	GET	-	?	Unknown	None
http://zero.webappsecurity.com/cgi.zip	GET	-	?	Unknown	None
http://zero.webappsecurity.com/test/cgi.zip	GET	-	?	Unknown	None
Severity	Check	Type filter criteria...	Method:Get		
Vulnerabilities	Not Found	Information	Best Practices	Scan Log	Server Information

Note that the filtering criteria (Method:Get) appears in the combo box, which also contains a red **X**. Click it to remove the filter and return the list to the original contents.

To specify multiple filters when typing criteria in the Filtering combo box, insert a comma between filters (such as Parameter:noteid, Method:GET).

Filter Criteria

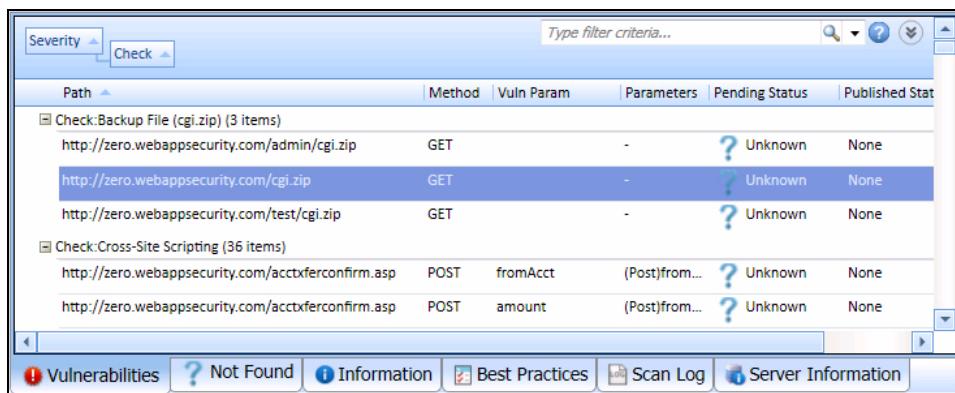
You can enter the following identifiers:

- check - Check name

- cookienamerp - Cookie name in the HTTP response
- cookienamerq - Cookie name in the HTTP request
- cookievaluerp - Cookie value in the HTTP response
- cookievaluerq - Cookie value in the HTTP request
- duplicates - Duplicates detected by SecurityScope
- filerq - File name and extension in the HTTP request
- headernamerp - Header name in the HTTP response
- headernamerq - Header name in the HTTP request
- headervaluerp - Header value in the HTTP response
- headervaluerq - Header value in the HTTP request
- location - Path plus parameters identifying the resource
- manual - A location added manually (syntax is manual:True or manual:False)
- method - HTTP method (GET, POST)
- methodrq - Method specified in HTTP request
- parameters - Parameters specified in the HTTP request
- path - Path identifying the resource (without parameters)
- rawrp - Raw HTTP response
- rawrq - Raw HTTP request
- sessiondataid - Session data identifier (right-click on a session in the Navigation pane and select Filter by Current Session)
- severity - Severity assigned to a vulnerability (critical, high, medium, low)
- stack - Stack trace returned by SecurityScope (syntax is stack:True or stack:False)
- statuscode - HTTP status code
- typerq - Type of request: query, post, or SOAP
- vparam - The vulnerability parameter

Using Groups

You can group items into categories based on the column headings. To do so, simply drag the heading and drop it on the grouping area at the top of the pane. Vulnerabilities in the following illustration are grouped by risk and then by check name.



The screenshot shows the WebInspect interface with a list of vulnerabilities. The columns are Path, Method, Vuln Param, Parameters, Pending Status, and Published Stat. The vulnerabilities are grouped under two main categories: 'Check:Backup File (cgi.zip) (3 items)' and 'Check:Cross-Site Scripting (36 items)'. Within each category, individual vulnerabilities are listed with their specific details like URL, method (GET or POST), parameters, and status.

If you right-click a column header, WebInspect displays the following shortcut menu:

- **Group by Field** - Groups vulnerabilities according to the field you selected.
- **Group by Box** - Shows the “Group By” area in which you can arrange grouping by column headers.
- **Columns** - Allows you to select which columns are displayed.
- **Save as Default View** - Saves the current grouping paradigm as the default for all scans.
- **Reset Default View** - Restores the grouping paradigm to the default view that you created.
- **Reset Factory Settings** - Restores the grouping paradigm to the original view (Severity > Check).

WebInspect Toolbars

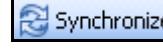
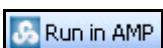
The *WebInspect* window contains two toolbars: Scan and Standard. You can display or hide either toolbar by selecting **View → Toolbars**.

The following table illustrates the buttons available on WebInspect’s Scan toolbar.

Scan Toolbar Buttons

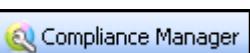
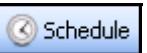
Button	Function
 Start / Resume	You can pause a scan and then resume scanning. Also, a completed scan may contain sessions that were not sent (because of timeouts or other errors); if you click Start , WebInspect will attempt to resend those sessions.
 Pause	Interrupts an ongoing scan. You can continue scanning by clicking the Start/Resume icon.

Scan Toolbar Buttons (cont'd)

Button	Function
 Skip	When conducting a sequential crawl and audit, you can skip processing by whichever engine is running (if you selected Test each engine type per session) or you can skip processing the session (if you selected Test each session per engine type). See Scan Settings: Method on page 165 for more information.
 Audit	If you conduct a crawl-only scan or a Step Mode scan, you can afterwards click this icon to conduct an audit.
 Rescan ▾	This button appears only if you select a tab containing a scan. If you select Scan Again from the drop-down menu, it launches the Scan Wizard prepopulated with settings last used for the selected scan. If you select Verify Vulnerabilities , it starts a scan that examines only those portions of the target site in which vulnerabilities were detected during the original scan. For more information, see Retesting/Reviewing Vulnerabilities on page 85.
 Compare	This button appears only if you select a tab containing a scan. It allows you to compare the vulnerabilities revealed by two different scans of the same target. For more information, see Compare Scans on page 157.
 Synchronize	This button appears only after connecting to WebInspect Enterprise. It allows you to specify a Software Security Center (SSC) project and version. WebInspect then downloads a list of vulnerabilities from SSC, compares the downloaded vulnerabilities to the vulnerabilities in the current scan, and assigns an appropriate status (New, Existing, Reintroduced, or Not Found) to the vulnerabilities in the current scan.
 Publish	This button appears only after connecting to WebInspect Enterprise and is enabled after you have synchronized WebInspect with Software Security Center. It uploads project version data through WebInspect Enterprise to Software Security Center..
 Run in AMP	This button appears only if WebInspect is connected to the Assessment Management Platform (AMP) and a scan is open on a tab. It allows you to send the scan settings to AMP, which creates a scan request and places it in the scan queue for the next available sensor. For detailed information, see Running a Scan in AMP or WebInspect Enterprise on page 109.
 Run in WebInspect Enterprise	This button appears only if WebInspect is connected to WebInspect Enterprise and a scan is open on the tab that has focus. It allows you to send the scan settings to WebInspect Enterprise, which creates a scan request and places it in the scan queue for the next available sensor.

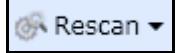
The following table illustrates the buttons available on WebInspect's Standard toolbar.

Standard Toolbar Buttons

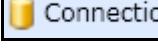
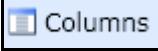
Button	Function
 New ▾	Allows you to start a Web site scan, a Web service scan, or an enterprise scan.
 Open ▾	Allows you to open a scan or a report.
 Compliance Manager	Opens the Compliance Manager, allowing you to create a compliance policy.
 Policy Manager	Opens the Policy Manager, allowing you to modify WebInspect scanning policies, and create custom checks to check for specific vulnerabilities.
 Report	If clicked when the Start page has focus, WebInspect prompts you for a scan and then allows you to select reports. If clicked when a scan tab has focus, prompts you to select reports for the open scan.
 Schedule	Allows you to schedule a scan to occur on a specific time and date.
 SmartUpdate	Contacts the central HP database to determine if updates are available for your system and, if updates exist, allows you to install them.
 AMP WebConsole	Launches the AMP Web Console application. This button appears only if you are connected to AMP.
 WebInspect Enterprise WebConsole	Launches the WebInspect Enterprise Web Console application. This button appears only if you are connected to WebInspect Enterprise.

The following table illustrates the buttons available on the Manage Scans toolbar.

Manage Scans Toolbar Buttons

Button	Function
 Open	To open scans, select one or more scans and click Open (or simply double-click an entry in the list). WebInspect loads the scan data and displays each scan on a separate tab.
 Rescan ▾	To launch the Scan Wizard prepopulated with settings last used for the selected scan, click Rescan → Scan Again . To rescan only those sessions that contained vulnerabilities revealed during a previous scan, select a scan and click Rescan → Retest Vulnerabilities . For more information, see Rescan the Site on page 157.

Manage Scans Toolbar Buttons (cont'd)

Button	Function
	To rename a selected scan, click Rename .
	To delete the selected scan(s), click Delete .
	To import a scan, click Import .
	To export a scan, export scan details, or export a scan to Software Security Center, click the drop-down arrow on Export .
	To compare scans, select two scans (using Ctrl + click) and click Compare . See Compare Scans on page 157 for more information.
	By default, WebInspect lists all scans saved in the local SQL Server Express Edition and in a configured SQL Server Standard Edition. To select one or both databases, or to specify a SQL Server connection, click Connections .
	When necessary, click Refresh to update the display.
	To select which columns should be displayed, click Columns . You can rearrange the order in which columns are displayed using the Move Up and Move Down buttons or, on the Manage Scans list, you can simply drag and drop the column headers.

WebInspect Menu Bar

The menu bar contains the following menus:

- File
- Scan
- Edit
- Enterprise Server
- View
- Reports
- Tools
- Help

File Menu

The following commands are available from the **File** menu.

New	Allows you to select either Web Site scan or Web Service scan, and then launches the Scan Wizard, which steps you through the process of starting a scan.
Open	Allows you to open either a scan or a report.
Schedule	Opens the <i>Manage Scan Scheduling</i> window, which allows you to add, edit, or delete a scheduled scan.
Import Scan	Allows you to import a scan file. Command appears only when a tab containing a scan is open.
Export	This command is available only when a tab containing a scan is selected. It allows you to export either a scan file, a scan details file, or a scan that is formatted for import into Software Security Center. For scan details, you can select one of the following (depending on the type of scan): comments, e-mails, full (all details), hidden fields, offsite links, parameters, requests, script, sessions, set cookies, URLs, vulnerabilities, web dump, or web forms.
Close Tab	When multiple tabs are open, closes the active tab.
Exit	Closes the WebInspect program.

Edit Menu

The following commands are available from the **Edit** menu.

Default Scan Settings	Displays the <i>Default Settings</i> window, allowing you to select or modify options used for scanning.
Current Scan Settings	Displays a settings window that allows you to select or modify options for the current scan. This command is available only when a tab containing a scan is selected.
Manage Settings	Opens a window that allows you to add, edit, or delete settings files.
Application Settings	Displays the <i>Application Settings</i> window, allowing you to select or modify options controlling the operation of the WebInspect application.
Copy URL	Copies the selected URL to the Windows clipboard. This command is available only when a tab containing a scan is selected.
Copy Scan Log	Copies the log (for the scan on the selected tab) to the Windows clipboard. This command is available only when a tab containing a scan is selected.

View Menu

The following commands are available from the **View** menu.

Word Wrap	Inserts soft returns at the right-side margins of the display area when viewing HTTP requests and responses. This command is available only when a tab containing a scan is selected.
Toolbars	Allows you to select which toolbars should be displayed.

Tools Menu

The **Tools** menu contains commands to launch the tool applications that are packaged with WebInspect.

Scan Menu

The **Scan** menu is visible only when a tab containing a scan has focus. It contains the following commands.

Start/Resume	Starts or resumes a scan after you paused the process.
Pause	Halts a crawl or audit. Click Resume to continue the scan.
Skip	If an audit is in progress, skips to the next audit methodology. If a crawl is in progress, skips to the audit.
Audit	Assesses the crawled site for vulnerabilities. Use the command after completing a crawl or exiting Step Mode.
Rescan	This command launches the Scan Wizard prepopulated with settings last used for the selected scan.

Enterprise Server Menu

The **Enterprise Server** menu contains the following commands. All commands except **Connect to AMP**, **Connect to WebInspect Enterprise**, and **About Enterprise Server** are available only when WebInspect is connected to one of the enterprise servers.

Connect to AMP/ Disconnect	Establishes or breaks a connection to the Assessment Management Platform (AMP) server.
Connect to WebInspect Enterprise/Disconnect	Establishes or breaks a connection to the WebInspect Enterprise (WIE) server.
Download Scan	Allows you to select a scan for copying from the server to your hard drive.

Publish Scan	Displays a dialog allowing you to review vulnerabilities and transmit them to an enterprise server which, in turn, transmits them to an HP Fortify Software Security Center server. For more information, see Publishing to Software Security Center on page 105. This command is not available until a connection is made to the enterprise server.
Upload Scan	Allows you to select a scan for transfer to an enterprise server. This is used most often when the application setting “auto upload scans” is not set.
Transfer Settings	Allows you to select a WebInspect settings file and transfer it to an enterprise server, which will create a Scan Template based on those settings. Also allows you to select an enterprise server Scan Template and transfer it to WebInspect, which will create a settings file based on the template. See Transferring Settings to or from an Enterprise Server on page 207 for more information.
WebConsole	Launches the AMP or WebInspect Enterprise Web Console application. This button appears only if you are connected to an enterprise server.
About Enterprise Server	Displays information about the Assessment Management Platform and WebInspect Enterprise.

Note: A WebInspect installation with a standalone license may connect to an enterprise server at any time, as long as the user is a member of a role in AMP or WebInspect Enterprise.

Reports Menu

The **Reports** menu contains the following commands.

Generate Report	Launches the Report Generator.
Manage Reports	Displays a list of standard and custom report types. You can rename, delete, or export custom-designed reports, and you may import a report definition file.
Report Designer	Launches the Report Designer, allowing you to define a custom report.

Help Menu

The following commands are available from the **Help** menu.

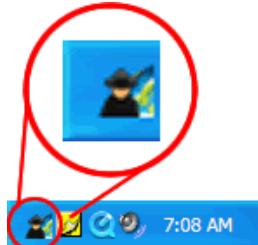
WebInspect Help	Opens the Help file.
Index	Opens the Help file, displaying the index in the left pane.
Search	Opens the Help file, displaying the search options in the left pane.

Support	<ul style="list-style-type: none"> • Request an Enhancement - If the support channel is enabled (see Support Channel on page 219), displays a window allowing you to submit enhancement requests to Hewlett-Packard. • Support Tool - Launches the HP Support Channel tool, which allows you to upload files that may help HP support personnel analyze and resolve any problems you encounter. • Technical Support - Displays instructions for contacting HP Technical Support.
Tutorials	Allows you to download tutorials and other WebInspect documentation.
About WebInspect	Displays information about the WebInspect application, including license information, allowed hosts, and attributes.

HP Application Security Center

The HP Application Security Center application, represented by an icon in the notification area of the taskbar, provides a context menu that allows you to:

- Start/stop the sensor service.
- Start/stop the scheduler service.
- Configure sensor.



Pop-up messages also appear whenever certain events occur.

This feature is provided primarily for users who install WebInspect as a standalone scanner, but subsequently want to connect to AMP or WebInspect Enterprise.

Inspecting Results: Web Site Scan

As soon as you start a Web Site Scan, WebInspect begins assessing your Web application and displays in the navigation pane an icon depicting each session (using either the Site or Sequence view). It also reports possible vulnerabilities on the **Vulnerabilities** tab and **Information** tab in the summary pane.

If you click a URL listed in the summary pane, the program highlights the related session in the navigation pane and displays its associated information in the information pane.

Sometimes the attack that detected a vulnerable session is not listed under attack information. That is, if you select a vulnerable session in the navigation pane and then click **Attack Info** in the **Session Info** panel, the attack information does not appear in the information

pane. This is because attack information is usually associated with the session in which the attack was created and not with the session in which it was detected. When this occurs, select the parent session and then click **Attack Info**.

If you right-click a vulnerability, a shortcut menu allows you to:

- **Copy URL** - Copies the URL to the Windows clipboard.
- **Copy Selected Item(s)** - Copies the text of selected items to the Windows clipboard.
- **Copy All Items** - Copies the text of all items to the Windows clipboard.
- **Export** - Copy to a CSV file.
- **View in Browser** - Renders the HTTP response in a browser.
- **Filter by Current Value** - Restricts the display of vulnerabilities to those that satisfy the criteria you select. For example, if you right-click on “Post” in the Method column and then select **Filter by Current Value**, the list displays only those vulnerabilities that were discovered by sending an HTTP request that used the Post method.

Note that the filter criterion is displayed in the combo box in the upper right corner of the summary pane. Alternatively, you can manually enter or select a filtering criterion using this combo box. For additional details and syntax rules, see [Using Filters and Groups in the Summary Pane](#) on page 88.

- **Change Severity** - Allows you to change the severity level.
- **Edit Vulnerability** - Displays the *Edit Vulnerabilities* dialog, allowing you to modify various vulnerability characteristics.
- **Review Vulnerability** - Allows you to retest the vulnerable session, mark it as a false positive, or send it to HP Quality Center or IBM Rational ClearQuest. For more information, see [Retesting/Reviewing Vulnerabilities](#) on page 85.
- **Mark as** - Flags the vulnerability as either a false positive (and allows you to add a note) or as ignored. In both cases, the vulnerability is removed from the list. You can view a list of all false positives by selecting False Positives in the Scan Info panel. You can view a list of false positives and ignored vulnerabilities by selecting Dashboard in the Scan Info panel, and then clicking the hyperlinked number of deleted items in the statistics column.

Note: You can recover “false positive” and “ignored” vulnerabilities. See [Recover Deleted Items](#) for details.

- **Send to** - Converts the vulnerability to a defect and adds it to the HP Quality Center or IBM Rational ClearQuest database.
- **Remove Location** - Removes the selected session from the navigation pane (both Site and Sequence views) and also removes any associated vulnerabilities.

Note: You can recover removed locations (sessions) and their associated vulnerabilities. See [Recovering Deleted Items](#) on page 85 for details.

- **Crawl** - Recrawls the selected URL.
- **Tools** - Presents a submenu of available tools.
- **Attachments** - Allows you to create a note associated with the selected session, flag the session for follow-up, add a vulnerability note, or add a vulnerability screenshot.

If you right-click a group, a shortcut menu allows you to:

- Collapse/Expand All Groups
- Collapse/Expand Group
- Copy Selected Item(s)

- Copy All Items
- Export
- Change Severity
- Mark as
- Send to
- Remove Location

Session

A session is a matched set comprising the HTTP request sent by WebInspect to test for vulnerabilities and the HTTP response received from the server. Each session employed during a crawl or scan is listed in the navigation pane.

Click a session in the navigation pane to view information about that session in the information pane.

Some views in the information pane contain controls at the top of the Information Display area that allow you to search the displayed text. To search using regular expressions, select the **Regex** check box. The status bar will display the line count, current line, and current column of a found item, as well as the total number of matched items.

Session Shortcut Menu

Right-clicking a session displays a shortcut menu allowing you to investigate the selected session. See [Navigation Pane Shortcut Menu](#) on page 64. The availability of commands depends on the session selected.

You can easily export to a text or XML file a list of every session URL, comments, hidden fields, and various other information. For help, see [Exporting Scans](#) on page 103.

Editing Vulnerabilities

After WebInspect assesses your application's vulnerabilities, you may want to edit and save the results for a variety of reasons, including:

- **Security**—If an HTTP request or response contains passwords, account numbers, or other sensitive data, you may want to delete or modify this information before making the scan results available to other persons in your organization.
- **Correction**—WebInspect occasionally reports a “false positive.” This occurs when WebInspect detects indications of a possible vulnerability, but further investigation by a developer determines that the problem does not actually exist. You can delete the vulnerability from the session or delete the entire session. Alternatively, you can designate it as a false positive (right-click the session in either the Site or Sequence view and select **Mark As False Positive**).
- **Severity Modification**—If you disagree with WebInspect’s ranking of a vulnerability, you can assign a different level, using the following scale:

0 - 9	Normal
10	Information

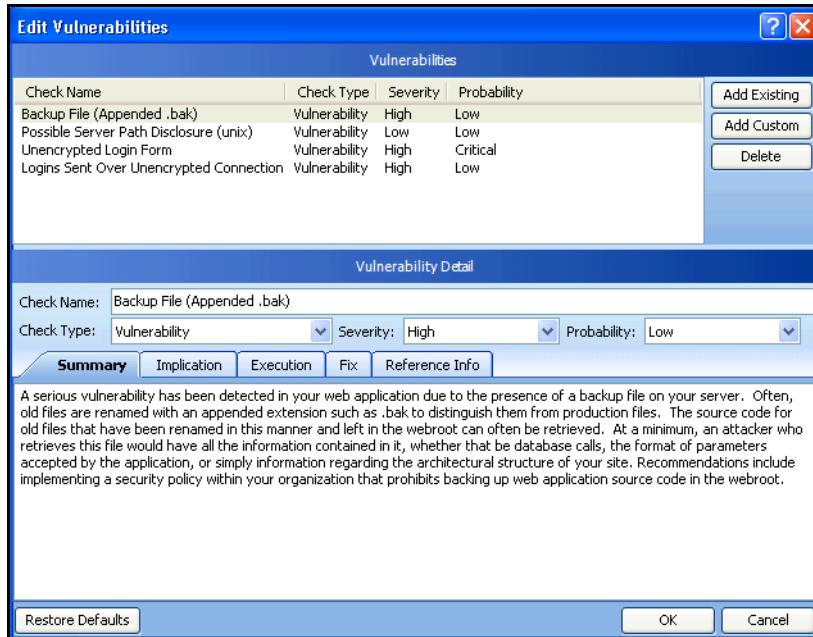
11 - 25	Low
26 - 50	Medium
51 - 75	High
76 - 100	Critical

- **Record Keeping**—You can modify any of the report fields associated with an individual vulnerability (Summary, Execution, Recommendation, Implementation, Fixes, and References). For example, you could add a paragraph to the Fixes section describing how you actually fixed the problem.
- **Enhancement**—If you discover a new vulnerability, you could define it and add it to a session as a custom vulnerability.

Follow the steps below to edit a session:

- 1 In the navigation pane, right-click a session containing a vulnerability
- or -
in the summary pane, right-click a URL.
- 2 Select **Edit Vulnerability** from the short-cut menu.

The *Edit Vulnerabilities* window displays.



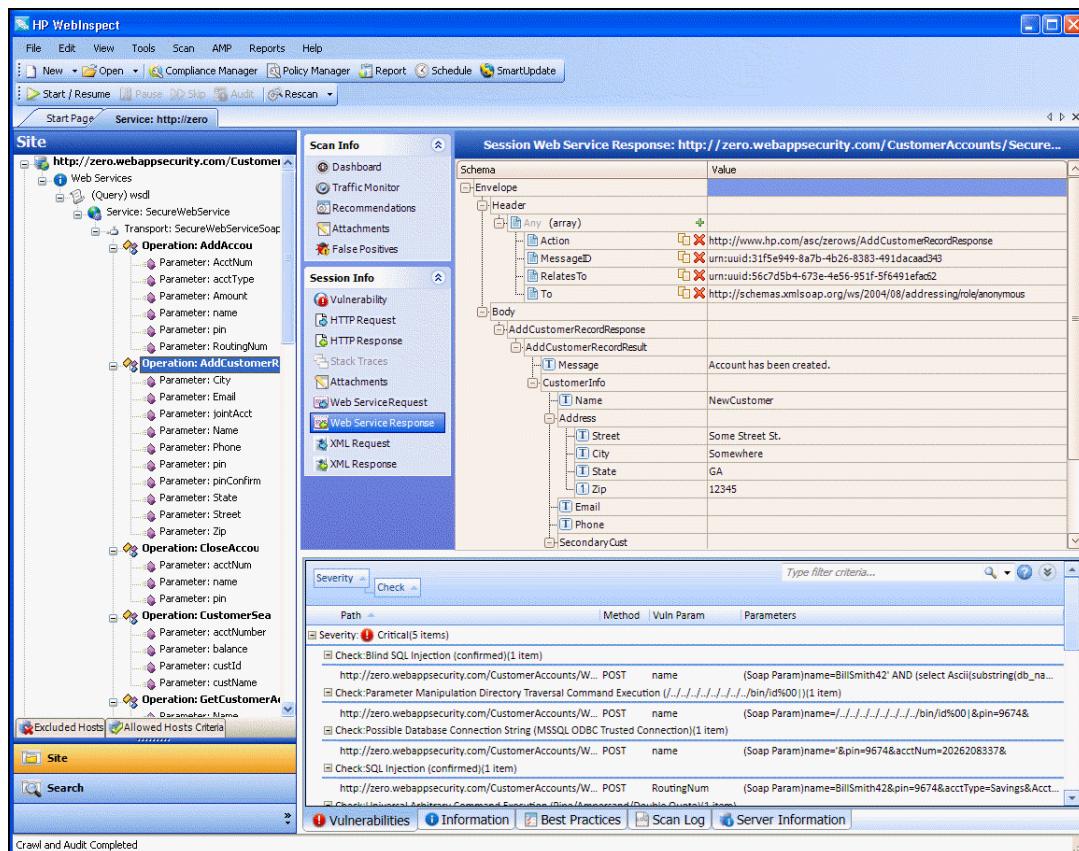
- 3 Select a vulnerability (if the session includes multiple vulnerabilities).
- 4 To add an existing vulnerability to the session (that is, one that exists in the database), click **Add Existing**.
 - a On the *Add Existing Vulnerability* window, enter part of a vulnerability name, or a complete vulnerability ID number or type.
 - b Click **Search**.
 - c Select one or more of the vulnerabilities returned by the search.
 - d Click **OK**.

- 5 To add a custom vulnerability, click **Add Custom**. You can then edit the vulnerability as described in Step 7.
- 6 To delete the vulnerability from the selected session, click **Delete**.
- 7 To modify the vulnerability, select different options from the Vulnerability Detail section. You can also change the descriptions that appear on the **Summary**, **Implication**, **Execution**, **Fix**, and **Reference Info** tabs.
- 8 Click **OK** to save the changes.

Inspecting Results: Web Service Scan

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. 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 understands how to communicate with the service. The WSDL document describes the procedures included in the Web service, the parameters those procedures expect, and the type of return information the client Web application will receive.



After selecting a session object in the navigation pane or on the **Vulnerabilities** tab of the summary pane, you can select options from the **Session Info** panel. See [Options in Session Info Panel](#) on page 76 for a description of those options.

Exporting Scans

WebInspect automatically saves scan results in the directories you specify in the Application settings.

Use the export function to transfer scan information to another WebInspect application or to Software Security Center.

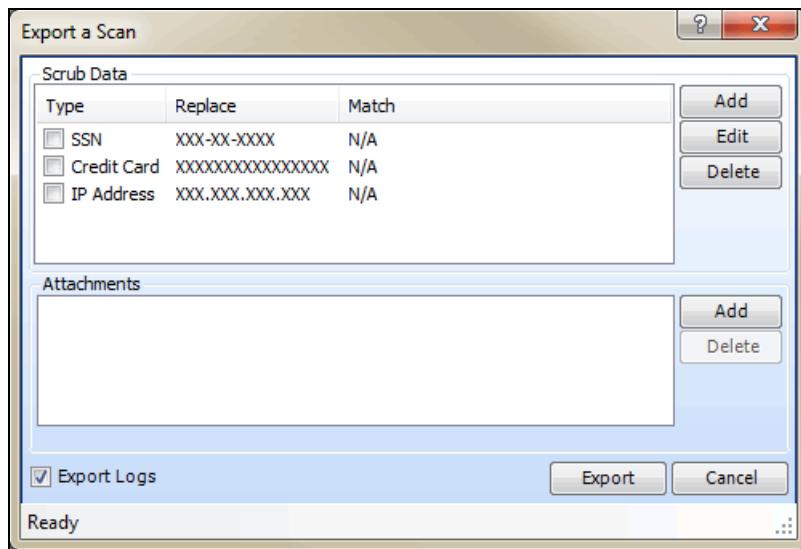
Follow the steps below to export a scan:

- 1 Open the scan you want to export (or click a tab containing an open scan) and click **File** → **Export** → **Scan** or **File** → **Export** → **Scan to Software Security Center**.

- or -

on the Manage Scans pane of the Start page, select a scan, click the drop-down arrow on the **Export** button and select either **Export Scan** or **Export Scan to Software Security Center**.

The *Export a Scan* window (or the *Export Scan to Software Security Center* window) appears.



- 2 The **Scrub Data** group contains, by default, three non-editable regular expression functions that will substitute an X for each digit in a string formatted as a Social Security number, credit card number, or IP address. To include a search-and-replace function, select its associated check box.
- 3 To create a scrubbing function:
 - a Click **Add**.
 - b On the *Add Scrub Entry* window, select either **Regex** or **Literal** from the **Type** list.
 - c In the **Match** box, enter the string (or a regular expression representing a string) that you want to locate. If using a regular expression, you can click the browse button [...] to open the Regular Expression Editor, with which you can create and test your regular expression.
 - d In the **Replace** box, enter the string that will replace the target specified by the Match string.

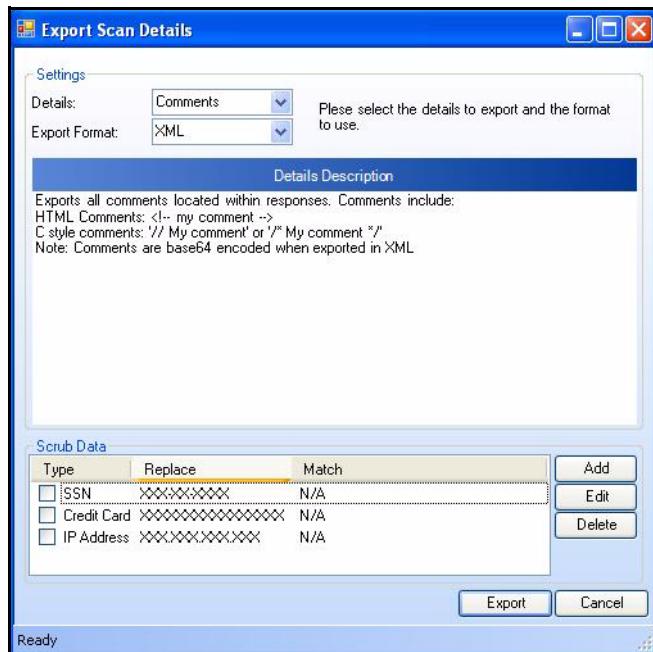
- e Click **OK**.
- Note: The *Export Scan to Software Security Center* window does not contain an option for including attachments or logs. If you are exporting to Software Security Center, go to Step 6.
- 4 If you are exporting to Software Security Center, go to Step 7.
 - 5 If you want to include an attachment:
 - a In the **Attachments** group, click **Add**.
 - b Using the standard file-selection window, navigate to the directory that contains the file you want to attach.
 - c Select a file and click **Open**.
 - 6 To include the scan's log files, select **Export Logs**.
 - 7 Click **Export**.
 - 8 Using the standard file-selection window, select a location and click **Save**.

Exporting Scan Details

Use the Export function to save information collected during a WebInspect crawl or audit.

- 1 Open a scan, or click a tab containing a scan.
- 2 Click **File → Export → Scan Details**.

The *Export Scan Details* window appears.



- 3 From the **Details** list, select the type of information you want to export. The available items depend on the type of scan conducted.
- 4 Choose a format (either Text or XML) from the **Export Format** list.

- 5 The **Scrub Data** group contains, by default, three non-editable regular expression functions that will substitute an X for each digit in a string formatted as a Social Security number, credit card number, or IP address. To include this search-and-replace function for a data type, select its associated check box.
- 6 To create a Scrub Data function:
 - a Click **Add**.
 - The *Add Scrub Entry* window appears.
 - b Select either **Regex** or **Literal** from the **Type** list.
 - c In the **Match** box, enter the string (or a regular expression representing a string) that you want to locate. If using a regular expression, you can click the browse button [...] to open the Regular Expression Editor, with which you can create and test your regular expression.
 - d In the **Replace** box, enter the string that will replace the target specified by the Match string.
 - e Click **OK**.
- 7 Click **Export**.
- 8 Using a standard file-selection window, enter a name for the exported file and click **Save**.

Publishing to Software Security Center

Publishing through WebInspect Enterprise

Use the following procedure to transmit scan data from WebInspect to an HP Fortify Software Security Center server, via WebInspect Enterprise.

Note: For information about managing the SSC status of vulnerabilities when conducting multiple scans of the same Web site or application, see [Integrating with WebInspect Enterprise](#) on page 106.

- 1 Configure WebInspect Enterprise and Software Security Center.
- 2 Run a scan in WebInspect (or use an imported or downloaded scan).
- 3 Click the **Enterprise Server** menu and select **Connect to WebInspect Enterprise**. You will be prompted to submit credentials.
- 4 If a scan is open on a tab that has focus, and you want to publish only that scan:
 - a Click  **Synchronize**.
 - b Select a project and version, then click **OK**.
 - c Examine the results. Columns will appear in the Summary pane specifying "Published Status" and "Pending Status." The Published Status is the status of the vulnerability the last time this scan was published to WebInspect Enterprise. The Pending Status is what the status of the vulnerability will be after this scan is published. Depending on the Pending Status, you can modify it to specify whether the vulnerability has been resolved or is still existing (see Step 7 below). In addition, a new tab named "Not Found" appears; this tab contains vulnerabilities that were

detected in previous scans but not in the current scan. You can add screenshots and comments to vulnerabilities or mark vulnerabilities as false positive or ignored. You can also review and retest vulnerabilities, modifying the scan results until you are ready to publish.

- d Click  Publish . Go to step 7.
- 5 To select from a list of scans:
 - a Click the **Enterprise Server** menu and select **Publish Scan**.
 - b On the *Publish Scan(s) to Software Security Center* dialog, select one or more scans.
 - c Select a project and project version.
 - d Click **Next**. WebInspect automatically synchronizes with SSC.
- 6 WebInspect lists the number of vulnerabilities to be published, categorized by status and severity. To determine the status, WebInspect compares previously submitted vulnerabilities (obtained by synchronizing with SSC) with those reported in the current scan. If this is the first scan submitted to a project version, all vulnerabilities will be "New."

If a vulnerability was previously reported, but is not in the current scan, it is marked as "Not Found." You must determine if it was not found because it has been fixed or because the scan was configured differently (for example, you may have used a different scan policy, or you scanned a different portion of the site, or you terminated the scan prematurely). When examining the results (step 4c), you can change the "pending status" of individual vulnerabilities detected by all but the first scan (by right-clicking a vulnerability in the Summary pane). However, when publishing, you must specify how WebInspect should handle any remaining "Not Found" vulnerabilities.

To retain these "Not Found" vulnerabilities in Software Security Center (indicating that they still exist), select **Retain: Assume all vulnerabilities still marked "Not Found" in the scan are still present**.

To remove them (implying that they have been fixed), select **Remove: Assume all vulnerabilities still marked "Not Found" in the scan are fixed**.

- 7 If this scan was conducted in response to a scan request initiated at HP Fortify Software Security Center, select **Associate scan with an "In Progress" scan request for the current project version**.
- 8 Click **Publish**.

Note: When you click **Publish**, the scan is queued for uploading to WebInspect Enterprise. After it is uploaded, WebInspect Enterprise will queue the scan for publishing to SSC. For this reason, the results of the publish in the Pending Status and Published Status columns will not be shown immediately. To see these new statuses after the publish has completed, the scan will need to be synchronized. To monitor the status of the upload and publish operation, you can connect to WebInspect Enterprise using the **WebInspect Enterprise WebConsole** button.

Integrating with WebInspect Enterprise

HP Fortify Software Security Center (SSC) is a suite of tightly integrated solutions for identifying, prioritizing, and fixing security vulnerabilities in software. It uses HP Fortify Static Code Analyzer to conduct static analysis and HP WebInspect to conduct dynamic application security testing. WebInspect Enterprise provides a central location for managing multiple WebInspect scanners and correlating scan results that can be published directly to individual project versions within SSC.

WebInspect Enterprise maintains a history of all vulnerabilities for a particular SSC project version. After WebInspect conducts a scan, it synchronizes with WebInspect Enterprise to obtain that history, compares vulnerabilities in the scan with those in the history, and then assigns a status to each vulnerability as follows:

SSC Status	Description
New	A previously unreported issue.
Existing	A vulnerability in the scan that is already in the history.
Not Found	A vulnerability in the history that is not found in the scan. This can occur because (a) the vulnerability has been remediated and no longer exists, or (b) because the latest scan used different settings, or scanned a different portion of the site, or for some other reason did not discover the vulnerability.
Resolved	A vulnerability that has been fixed.
Reintroduced	A vulnerability that appears in a current scan but was previously reported as "Resolved."
Still an Issue	A vulnerability that was "Not Found" in the current scan does, in fact, exist.

To change the SSC status for an individual vulnerability, right-click a vulnerability on the **Vulnerability** tab and select **Modify Publish Status**. This option appears only after connecting to WebInspect Enterprise and is enabled only after you have synchronized WebInspect with Software Security Center.

The following example demonstrates a hypothetical series of scans for integrating vulnerabilities into HP Fortify Software Security Center.

First scan

- 1 Scan the target site with WebInspect. In this example, assume that only one vulnerability (Vuln A) is discovered.
- 2 Examine the results. You can add screenshots and comments to vulnerabilities or mark vulnerabilities as false positive or ignored. You can also review, retest, and delete vulnerabilities.
- 3 Synchronize the scan with a project version in SSC, then publish the scan.

Second scan

- 1 The second scan again reveals Vuln A, but also discovers four more vulnerabilities (Vulns B, C, D, and E).
- 2 Synchronize the scan with the project version in SSC.
- 3 Now examine the results. If you added audit data (such as comments and screenshots) to Vuln A when publishing the first scan, the data will be imported into the new scan.
- 4 Publish the scan to SSC. Vuln A will be marked "Existing," Vulns B-E will be marked "New," and five items will exist in the SSC system.

Third scan

- 1 The third scan discovers Vulns B, C, and D, but not Vuln A or Vuln E.
- 2 Synchronize the scan with the project version in SSC.
- 3 After retesting Vuln A, you determine that it does, in fact, exist. You change its pending status to “Still an Issue.”
- 4 After retesting Vuln E, you determine that does not exist. You change its pending status to “Resolved.”
- 5 Publish the scan to SSC. Vulns B, C, and D will be marked “Existing.” Five items will exist in the SSC system.

Fourth Scan

- 1 The fourth scan does not find Vuln A or Vuln B. The scan does find Vulns C, D, E, and F.
- 2 Synchronize the scan with the project version in SSC.
- 3 Vuln E was previously declared to be resolved and so its status is set to “Reintroduced.”
- 4 You examine the vulnerabilities that were not found (A and B, in this example). If you determine that the vulnerability still exists, update the pending status to “Still an Issue.” If a retest verifies that the vulnerability does not exist, update the pending status to “Resolved.”
- 5 Publish the scan to SSC. Vulns C and D remain marked “Existing.”

Publishing through AMP

Use the following procedure to publish scan data to HP Fortify Software Security Center when connected to AMP:

- 1 Click the WebInspect **Enterprise Server** menu and select **Publish Scan**.
- 2 On the *Publish Scan(s) to Software Security Center* window, select a WebInspect scan from the Scan Name column.

Note: To access scans in a different database, click **Connections** and, in the Database application settings, change options under **Connection Settings for Scan Viewing**.

- 3 Select a site from the drop-down list in the **AMP** group. You can view organizations and projects by selecting **Show Organizations**.

Each scan must be assigned to a site. The program attempts to select the correct site based on the “Scan URL” in the scan file, but you may select an alternative.

If an appropriate site does not exist, you can create a site in AMP using the following procedure. You must have AMP permission to create a site.

- a Click **Create Site** (at the top right of the dialog).
- b On the *Create Site on AMP Server* window, provide the following information.
 - Project: Select a project name.
 - Site Name: Enter a name that identifies this site.
 - URL: Enter a fully qualified domain name or an IP address.
 - Phase: (Optional) Enter the name of a phase or select an existing name from the Phase list. If you assign phases to sites, you can display only those sites that are members of a specific phase.

- Group: (Optional) Enter the name of a group or select an existing name from the Group list. If you create groups of sites, you can display only those sites that are members of a specific group.
 - Authentication: If authentication is required, select a type from the list.
 - Weight: Weight is used to calculate the risk score that appears on the Sites form. The risk score for a site is equal to the risk score of the most recently completed scan of that site multiplied by the value you select here. It allows the user to indicate that some sites are more important or have a higher risk than others. For example, the risk associated with vulnerabilities in an external-facing site could be weighted higher than vulnerabilities in an internal-only site because of the level of exposure. The weight can be any value between zero and 10.
 - c Click **OK**.
- 4 In the **Software Security Center** group, click **Login**.
 - 5 Enter your credentials and click **OK**.
 - 6 Select a project version.
 - 7 Click **Publish**.

Importing Scans

Follow the steps below to import a scan that was conducted by another instance of WebInspect.

- 1 Click **File** → **Import Scan**.
- 2 Using a standard file-selection window, select an option from the **Files of Type** list:
 - **Scan files (*.scan)**—scan files designed for or created by WebInspect versions beginning with 7.0.
 - **SPA files (*.spa)**—scan files created by versions of WebInspect prior to release 7.0.
- 3 Choose a file and click **Open**.

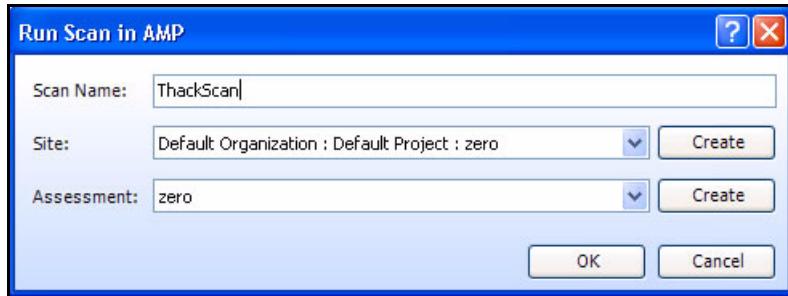
If attachments were exported with the scan, those attachments will be imported and saved in a subdirectory of the imported scan. The default location is C:\Documents and Settings\<username>\Local Settings\Application Data\HP\HP WebInspect\ScanData\Imports\<DirectoryName>\<filename>, whereDirectoryName is the ID number of the exported/imported scan.

Running a Scan in AMP or WebInspect Enterprise

This feature is designed for users who prefer to configure a scan in WebInspect rather than AMP or WebInspect Enterprise. You can modify the settings and run the scan in WebInspect, repeating the process until you achieve what you believe to be the optimal settings. You can then send the open scan's settings to AMP or WebInspect Enterprise, which creates a scan request and places it in the scan queue for the next available sensor.

- 1 Open a scan.

- 2 If you are not connected to an enterprise server, click the **Enterprise Server** menu and select **Connect to AMP** or **Connect to WebInspect Enterprise**.
- 3 Click the **Scan** menu and select **Run in AMP** or **Run in WebInspect Enterprise** (or simply click the appropriate button on the toolbar).
- 4 If you are connected to AMP:
 - a On the *Run Scan in AMP* window, type a name in the **Scan Name** box, or leave the name blank to allow AMP to assign a name.



- b If a site exists on AMP with a URL corresponding to the scan, that site is selected by default. You can change the site or you can create a site, using the appropriate buttons.

Note: When you connect to AMP 8.1 or greater, the **Site** list displays the organization and project names in addition to the site name. For AMP 8.0 and earlier, only the site name is listed.

- c (Optional) Select an assessment with which this scan will be associated, or click **Create** to create an assessment.

Note: Assessments are enabled only if WebInspect is connected to AMP version 9.0 or later. Also, to create an assessment, you must have administrative permission within AMP.

- d Click **OK**.
- For AMP 8.0 or earlier, you must have permission to create custom scans. For AMP 8.1 or later, you must have permission to create scans in the project.
- 5 If you are connected to WebInspect Enterprise:
 - a On the *Run Scan in WebInspect Enterprise* dialog, enter a name for the scan.



- b Select a project and a project version.
 - c Click **OK**.
- If you pass all permission checks, the scan is created and the priority assigned to the scan is the highest priority allowed by your role (up to 3, which is the default).

Uploading a Scan to an Enterprise Server

Use the following procedure to upload a scan file from WebInspect to an enterprise server (Assessment Management Platform or WebInspect Enterprise).

- 1 Click the **WebInspect Enterprise Server** menu and select **Upload Scan**.
- 2 On the *Upload Scan(s)* window, select one or more WebInspect scans from the Scan Name column. To access scans in a different database, click **Connections** and, in the Database application settings, change options under **Connection Settings for Scan Viewing**.
- 3 If uploading to AMP, then for each scan, select a site from the drop-down list in the AMP Site column.

Each scan must be assigned to a site. The program attempts to select the correct site based on the "Scan URL" in the scan file, but you may select an alternative. If an appropriate site does not exist, you can create one (if you have AMP permission to create a site).

To create a site:

- a Click **Create Site**.
 - b On the *Create Site on AMP Server* window, provide the following information.
 - Project: Select a project name.
 - Site Name: Enter a name that identifies this site.
 - URL: Enter a fully qualified domain name or an IP address.
 - Phase: (Optional) Enter the name of a phase or select an existing name from the Phase list. If you assign phases to sites, you can display only those sites that are members of a specific phase.
 - Group: (Optional) Enter the name of a group or select an existing name from the Group list. If you create groups of sites, you can display only those sites that are members of a specific group.
 - Authentication: If authentication is required, select a type from the list.
 - Weight: Weight is used to calculate the risk score that appears on the Sites form. The risk score for a site is equal to the risk score of the most recently completed scan of that site multiplied by the value you select here. It allows the user to indicate that some sites are more important or have a higher risk than others. For example, the risk associated with vulnerabilities in an external-facing site could be weighted higher than vulnerabilities in an internal-only site because of the level of exposure. The weight can be any value between zero and 10.
 - c Click **OK**.
 - 4 (Optional, if uploading to AMP) Select an assessment from the drop-down list in the Assessment column.
- If an appropriate assessment does not exist, you can create an assessment in AMP using the following procedure. You must have AMP permission to create an assessment. Note that this feature is enabled and relevant only when WebInspect is connected to AMP 9.0 or above.
- a Click **Create Assessment**.
 - b Select a site with which this assessment will be associated.
 - c Enter a name for the assessment and (optionally) a description.

- d Click **OK**.
- 5 If uploading to WebInspect Enterprise, then for each scan, select a project and project version from the appropriate drop-down lists.
The program attempts to select the correct project and project version based on the "Scan URL" in the scan file, but you may select an alternative.
- 6 Click **Upload**.
The **Show Organizations and Projects** check box is enabled only when WebInspect is connected to AMP 8.1 and above.

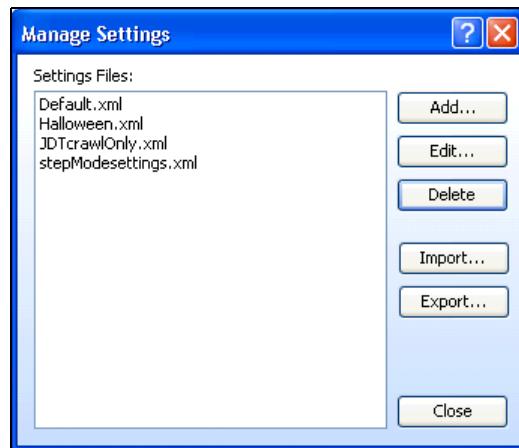
Managing Settings

This feature allows you to create, edit, delete, import, and export scan settings files.

Note: You can also load and save settings and restore factory default settings from the Default Settings window. Click **Edit** → **Default Scan Settings**.

- 1 From the WebInspect **Edit** menu, select **Manage Settings**.

The *Manage Settings* window opens.



- 2 To create a settings file:
 - a Click **Add**.
 - b On the *Create New Settings* window, change settings.
 - c When finished, click **OK**.
 - d Using a standard file-selection dialog, name and save the file.
- 3 To edit a settings file.
 - a Select a file.
 - b Click **Edit**.
 - c On the *Create New Settings* window, change settings.
 - d When finished, click **OK**.
- 4 To delete a settings file:
 - a Select a file.

- b Click **Delete**.
- 5 To import a settings file:
 - a Click **Import**.
 - b Using a standard file-selection dialog, select a settings file and click **Open**.
- 6 To export a settings file:
 - a Select a file.
 - b Using a standard file-selection dialog, select a settings file and click **Save**.

To scan with a saved settings file:

- 1 From the WebInspect **Edit** menu, select **Default Settings**.
- 2 At the bottom of the *Default Settings* window, in the left column, click **Load settings from file**.
- 3 Using a standard file-selection dialog, select the settings file you want to use and click **Open**.

The file you select is now your default settings file.

Managing Scans

This feature allows you to open, rename, or delete files containing the results of a previous scan. You can also choose a database: either Local (if scan files are stored in a SQL Server 2005 Express Edition database on your machine) or Remote (if scan files are stored in a SQL Server 2005 database, if configured), or both.

On the **Start Page** tab, click **Manage Scans**



A list of scans appears in the right-hand pane of the Start Page.

By default, WebInspect lists all scans saved in the SQL Server Express Edition on your machine and in SQL Server Standard Edition (if configured). The current state of the scan is indicated in the Status column.

(Optional) If you like, you can group scans into categories based on the column headings. To do so, simply drag the heading and drop it on the grouping area.

For descriptions of toolbar buttons and their functions, see [Manage Scans Toolbar Buttons](#) on page 92.

➤ You can also perform most of these functions, plus generate a report, by right-clicking an entry and selecting a command from the shortcut menu.

Managing Scheduled Scans

You can instruct WebInspect to conduct a scan at a time and date you specify. The options and settings you select are saved in a special file and accessed by a Windows service that starts WebInspect (if necessary) and initiates the scan. It is not necessary for WebInspect to be running at the time you designate the scan to begin.



Note: Scheduled scans, when complete, do not appear in the Recent Scans list that displays on the WebInspect **Start** page. To access scheduled scans after they are complete, select the **Start** page and click **Manage Scans**.

- 1 On the **Start Page** tab, click **Manage Schedule**



WebInspect lists all scheduled scans, including ones that have already run.

- 2 (Optional) If you like, you can group scheduled scans into categories based on the column headings. To do so, simply drag the heading and drop it on the grouping area.
- 3 Use the toolbar buttons to perform the functions listed below.



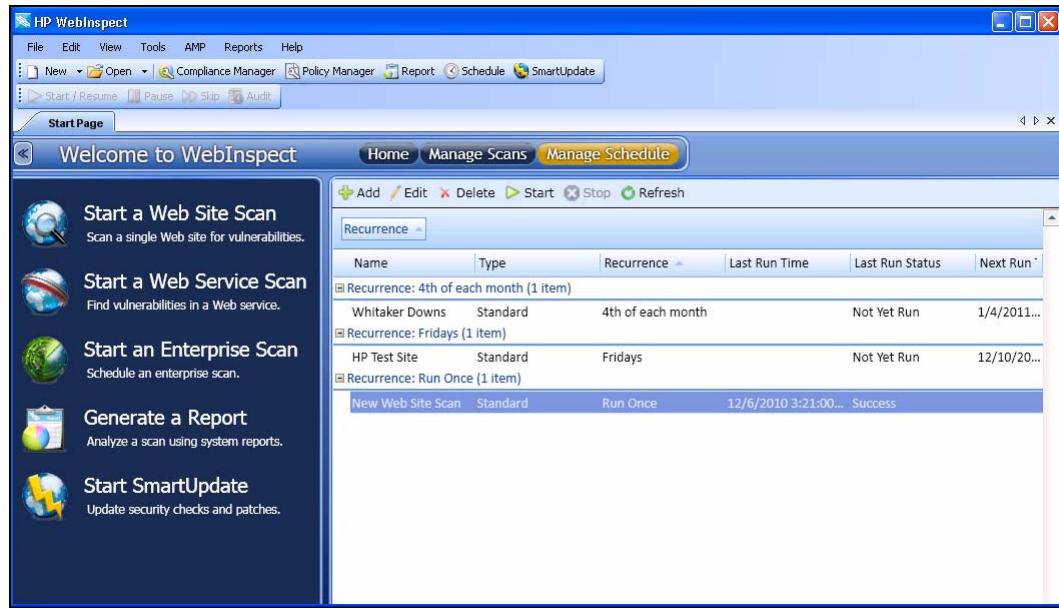
Button	Description
Add	To open the <i>Scheduled Scan Wizard</i> and schedule a scan, click Add .
Edit	To open the <i>Scheduled Scan Wizard</i> and edit settings for a scheduled scan, click Edit .
Delete	To delete a scan from the list, select a scan and click Delete .
Start	To run a scan immediately, without waiting for the scheduled time, select a scheduled scan and click Start (or right-click a scan and select Start Scan from the shortcut menu). As with all scheduled scans, the scan runs in the background and does not appear on a tab.
Stop	To stop a scheduled scan, select a scan that is running and click Stop (or right-click a running scan and select Stop Scan from the shortcut menu).
Refresh	When necessary, click Refresh to update the display.

Scheduling a Scan

Use the following procedure to schedule a scan.

- 1 Do one of the following:
 - On the WebInspect **Start** page, click **Manage Schedule**.
 - Click **File** → **Schedule**.

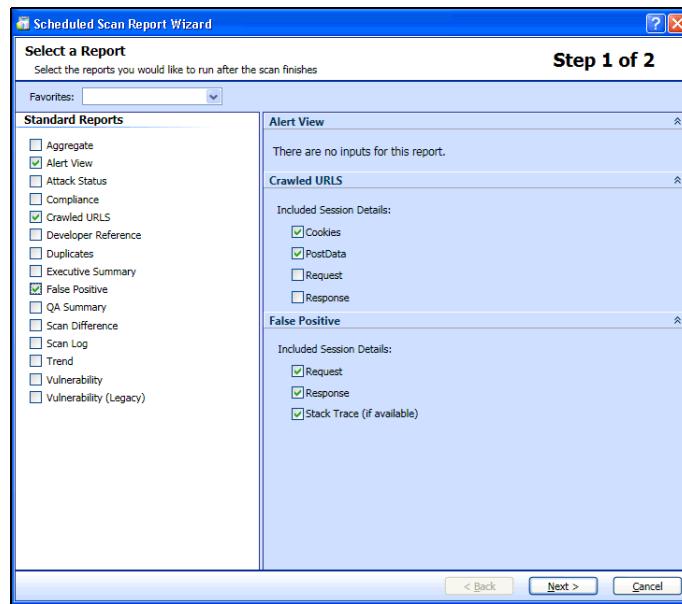
A list of scheduled scans appears in the right pane of the **Start** page.



- 2 Click **Add**.
- 3 In the Type of Scan group, choose one of the following:
 - Web Site Scan
 - Web Service Scan
 - Enterprise Scan
- 4 Specify when you want to conduct the scan. The choices are:
 - **Immediately**
 - **Run Once:** Modify the date and time when the scan should begin. You can click the drop-down arrow to reveal a calendar for selecting the date.
 - **Recurrence Schedule:** Use the slider to select a frequency (Daily, Weekly, or Monthly). Then specify the time when the scan should begin and (for Weekly or Monthly) provide other schedule information.
- 5 Click **Next**.
- 6 Enter the settings for the type of scan you selected.
- 7 For Web Site and Web Service Scans only, you can elect to run a report at the conclusion of the scan:
 - Select **Generate Reports** and click the Select Reports hyperlink.
 - Continue with Selecting a Report (below).
- 8 To schedule the scan without generating a report, click **Schedule**.

Selecting a Report

If you opted to include a report with the scheduled scan, the *Select a Report* dialog appears:



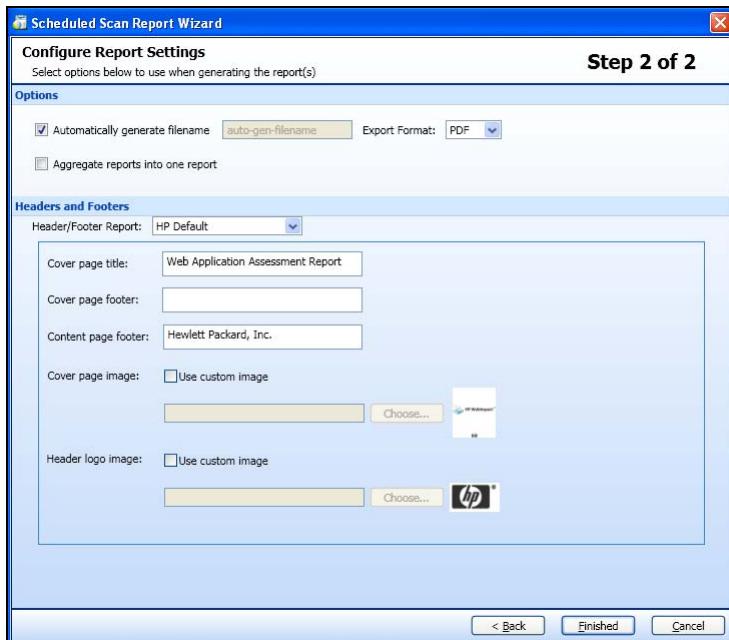
- 1 (Optional) Select a report from the **Favorites** list.

A “favorite” is simply a named collection of one or more reports and their associated parameters. To create a favorite once you have selected reports and parameters, click the **Favorites** list and select **Add to favorites**.

- 2 Select one or more reports.
- 3 Provide information for any parameters that may be requested. Required parameters are outlined in red.
- 4 Click **Next**.

The *Configure Report Settings* dialog appears.

Configure Report Settings



- 1 If you select **Automatically Generate Filename**, the name of the report file will be formatted as <reportname> <date/time>. <extension>. For example, if creating a compliance report in pdf format and the report is generated at 6:30 on April 5, the file name would be "Compliance Report 04_05_2009 06_30.pdf." This is useful for recurring scans.
Reports are written to the directory specified for generated reports in the Application settings.
- 2 If you did not select **Automatically Generate Filename**, enter a name for the file in the **Filename** box.
- 3 Select the report format from the **Export Format** list.
- 4 If you selected multiple reports, you can combine them all into one report by selecting **Aggregate reports into one report**.
- 5 Select a template that defines the headers and footers used for the report and, if necessary, provide the requested parameters.
- 6 Click **Finished**.
- 7 Click **Schedule**.

Stopping a Scheduled Scan

To halt a scheduled scan while it is running, select the scan from the Manage Schedule list and click (or right-click the scan and select **Stop Scan** from the shortcut menu).

To restart a stopped scan, select the scan from the Manage Schedule list and click (or right-click the scan and select **Start Scan** from the shortcut menu).

Generating Reports

Follow the steps below to generate a report for scans that have run. This procedure does not apply to scheduled scans.

- 1 Click the Report icon on the WebInspect toolbar
-or-
Select **Reports** → **Generate Report**
-or-
on the WebInspect **Start** page, click **Generate a Report**.

The *Generate a Report* window appears.

- 2 Select one or more scans (designated by name, URL, or IP address).
- 3 (Optional) Click **Advanced** (at the bottom of the window) to select options for saving reports and for selecting a template for headers and footers. See [Advanced Report Options](#) on page 120 for more information.
- 4 Click **Next**.
- 5 (Optional) Select a report from the **Favorites** list.

“Favorites” is simply a named collection of one or more reports and their associated parameters. To create a favorite once you have selected reports and parameters, click the **Favorites** list and select **Add to favorites**.

- 6 Select one or more reports. The following standard reports are available:
 - **Aggregate**—This report is designed for multiple scans. You can select which severity categories to report, report sections (server content and vulnerability detail), and session information (responses and requests). Stack traces can also be reported, when available.
 - **Alert View**—This report lists all vulnerabilities sorted by severity, with a hyperlink to each HTTP request that elicited the vulnerability. It also includes an appendix that describes each vulnerability in detail.
 - **Attack Status**—For each attack agent (check) employed during the scan, this report lists the vulnerability ID number, check name, vulnerability severity, whether or not the check was enabled for the scan, whether or not the check passed or failed (i.e., did or did not detect the vulnerability), and (if it failed) the number of URLs where the vulnerability was detected. You can select to report vulnerabilities of a certain severity as well as the pass/fail status.
 - **Compliance**—This report provides a qualitative analysis by grading how well your application complies with certain government-mandated regulations or corporate-defined guidelines. You must specify a compliance template. You can select a default template from the list or open the Compliance Manager and create a custom template. See [Compliance Templates](#) on page 121 for a description of available templates.

- **Crawled URLs**—For each URL encountered during the crawl, this report lists any cookies sent and the raw HTTP request and response.
- Note: If WebInspect is unable to complete this report or if the report is extremely slow to generate, modify the report as follows:
- 1 Open Report Designer.
 - 2 Open the Crawled URLs - Non-unique Subreport.
 - 3 For the RichTextBoxes underneath GroupHeaderRequest and GroupHeaderResponse, set the MaxLength property to 4096.
 - 4 Save the report.
- **Developer Reference**—This report presents totals and detailed descriptions of each form, JavaScript, e-mail, comment, hidden control, and cookie discovered on the Web site. You can select one or more of these reference types.
 - **Duplicates**—This report contains information about vulnerabilities detected by SecurityScope that were traceable to the same source. It begins with a bar chart comparing the total number of uncorrelated vulnerabilities to the number of unique vulnerabilities.
 - **Executive Summary**—This report lists basic statistics, plus charts and graphs that reflect your application's level of vulnerability.
 - **False Positives**—This report displays information about URLs that WebInspect originally classified as vulnerabilities, but were subsequently determined by a user to be false positives.
 - **QA Summary**—This report lists the URLs of all pages containing broken links, server errors, **external links, and timeouts**. You can select one or more of these categories.
 - **Scan Difference**—This report compares two scans and reports the differences, such as vulnerabilities, pages, and file-not-found responses that occur in one Web site but not the other.
 - **Scan Log**—Sequential list of the activities conducted by WebInspect during the scan (as the information appears on the **Scan Log** tab of the summary pane).
 - **Trend**—This report allows you to monitor your development team's progress toward resolving vulnerabilities. For example, you save the results of your initial scan and your team begins fixing the problems. Then once a week, you rescan the site and archive the results. To quantify your progress, you run a trend report that analyzes the results of all scans conducted to date. The report includes a graph showing the number of vulnerabilities, by severity, plotted on a timeline defined by the date on which each scan was conducted. Important: To obtain reliable results, make sure you conduct each scan using the same policy.
 - **Vulnerability (Legacy)**—This is a detailed report of each vulnerability, with recommendations concerning remediation.
 - **Vulnerability**—This report also presents detailed information about discovered vulnerabilities, sorted by severity.
- 5 Provide information for any parameters that may be requested. An exclamation mark ! indicates a required parameter.
 - 6 If you want to produce individual reports on separate tabs (rather than combining all reports on one tab), select **Open Reports in Separate Tabs**.
 - 7 Click **Finish**.

After WebInspect generates the report and displays it on a tab, you can save the report by clicking **Save As** on the Report Viewer toolbar.

Reports can be saved in the following formats:

- Adobe Portable Data Format (.pdf)
- Hypertext Markup Language (.html)
- Native WebInspect internal format (.raw)
- Rich Text Format (.rtf)
- Text (.txt)
- Microsoft Excel (.xls)

Advanced Report Options

Click **Advanced** (at the bottom of the *Generate a Report* window) to select options for saving reports and for selecting a template for headers and footers.



Save reports to disk

Select this option to output a report to a file.

Automatically generate file name

If you select this option when saving the report to disk, the name of the report file will be formatted as <reportname> <date/time>. <extension>. For example, if creating a compliance report in pdf format and the report is generated at 6:30 on April 5, the file name would be "Compliance Report 04_05_2009 06_30.pdf." This is useful for recurring scans.

If you select more than one report type, then <reportname> will be "Combined Reports."

Reports are written to the directory specified for generated reports in the Application settings.

If you do not select **Automatically generate filename**, replace the default name "auto-gen-filename" with a file name.

Export Format

Select a report format.

Headers and Footers

Select a format for the report's header and footer, and then enter or select the components.

Compliance Templates

The available compliance templates are described below. Additional templates may be downloaded through SmartUpdate as they become available.

21CFR11

Part 11 of Title 21 of the United States Code of Federal Regulation (commonly abbreviated as “21 CFR 11”) includes requirements for electronic records and electronic signatures. To assist medical companies in compliance, the US Food and Drug Administration (FDA) has published guidance for the proper use of electronic records and electronic signatures for records that are required to be kept and maintained by FDA regulations. The guidance outlines “criteria under which the agency considers electronic records, electronic signatures, and handwritten signatures executed to electronic records to be trustworthy, reliable, and generally equivalent to paper records and handwritten signatures executed on paper.”

Due to the law and FDA guidance, medical companies and organizations dealing with highly sensitive medical information are being required to ensure that electronic records and electronic signatures are trustworthy, reliable, and generally an equivalent substitute for paper records and handwritten signatures. As interaction between equipment, operators, and computers becomes commonplace, it is important to establish a secure means to communicate and store information.

Basel II

Basel II is a round of deliberations by central bankers from around the world, under the auspices of the Basel Committee on Banking Supervision (BCBS) in Basel, Switzerland, aimed at producing uniformity in the way banks and banking regulators approach risk management across national borders. The BCBS is the international rule-making body for banking compliance. In 2004, central bank governors and the heads of bank supervisory authorities in the Group of Ten (G10) countries endorsed the publication of “International Convergence of Capital Measurement and Capital Standards: a Revised Framework,” the new capital adequacy framework commonly known as Basel II.

Basel II essentially requires banks to increase their capital reserves or demonstrate that they can systematically and effectively control their credit and operational risk. The framework defines operational risk as “the risk of loss resulting from inadequate or failed internal processes, people and systems or from external events,” and highlights hacking and information theft through inadequate systems security as loss events. While banks around the world are experts at managing risk by virtue of operating in global financial markets, they are relatively new at understanding and controlling the risks inherent with operating online banking systems and keeping customer data secure.

Banks that practice effective information and systems security are able to demonstrate to regulators that they should qualify for lower capital reserves through reduced operational risk. The Basel II framework insists that banks demonstrate that an effective system of policies and processes are in place to protect information and that compliance to these policies

and processes is ensured, but is not prescriptive in how banks should implement security policies and processes. The international standard ISO/IEC 17799 Code of Practice for Information Security Management provides guidelines for implementing and maintaining information security and is commonly used as a model for managing and reporting operational risk related to information security in the context of Basel II.

CA OPPA

The California Online Privacy Protection Act (OPPA) was established in 2003 to require all businesses and owners of commercial web sites in the state of California to conspicuously post and comply with a privacy policy that clearly states the policies on the collection, use, and sharing of personal information. The policy identifies the categories of personally identifiable information collected about site visitors and the categories of third parties with whom the operator may share the information.

Any business, organization, or individual that operates a Web site that collects private personal information for a person residing in the state of California is bound by the provisions of the law, so the California OPPA has a much greater impact nationally than is typical for state legislation.

CASB 1386

California Senate Bill 1386 has established the most specific and restrictive privacy breach reporting requirements of any state in the United States. The law was enacted to force businesses, organizations, and individuals holding private personal information for legitimate business purposes to inform consumers immediately when their personal information has been compromised. The law also gives consumers the right to sue businesses in civil court for damages incurred through the compromise of information. Any business, organization, or individual that holds private personal information for a person residing in the state of California is bound by the provisions of the law.

COPPA

The Children's Online Privacy Protection Act (COPPA) was enacted in 2000 to protect the online collection of personal information about children under the age of 13. COPPA's goal was to protect children's privacy and safety online in recognition of the easy access that children often have to the Web. The law requires that Web site operators post a privacy policy on the site and outlines requirements for Web site operators to seek parental consent to collect children's personal information in certain circumstances.

The law applies not only to Web sites that are clearly directed toward children but to any Web site that contains general audience content where the Web site operators have actual knowledge that they are collecting personal information from children. An operator must post a link to a notice of its information practices on the home page of its Web site or online service and at each area where it collects personal information from children. An operator of a general audience site with a separate children's area must post a link to its notice on the home page of the children's area.

DCID

This directive establishes the security policy and procedures for storing, processing, and communicating classified intelligence information in information systems. For purposes of this directive, intelligence information refers to sensitive compartmented information and special access programs for intelligence under the purview of the Director of Central Intelligence.

DoD Application Security Checklist Version 2

DISA Field Security Operations (FSO) conducts Application SRRs to provide a minimum level of assurance to DISA, Joint Commands, and other Department of Defense (DoD) organizations that their applications are reasonably secure against attacks that would threaten their mission. The complexity of most mission critical applications precludes a comprehensive security review of all possible security functions and vulnerabilities in the time frame allotted for an Application SRR. Nonetheless, the SRR helps organizations address the most common application vulnerabilities and identify information assurance (IA) issues that pose an unacceptable risk to operations.

Ideally, IA controls are integrated throughout all phases of the development life cycle. Integrating the Application Review process into the development lifecycle will help to ensure the security, quality, and resilience of an application. Since the Application SRR is usually performed close to or after the applications release, many of the Application SRR findings must be fixed through patches or modifications to the application infrastructure. Some vulnerabilities may require significant application changes to correct. The earlier the Application Review process is integrated into the development life cycle, the less disruptive the remediation process will be.

DoD Application Security and Development STIG V3 R2

This compliance template reports all applicable web application components of the Application Security and Development Security Technical Implementation Guide (STIG) Version 3, Release 1. The STIG provides security guidance for use throughout the application development lifecycle. Defense Information Systems Agency (DISA) encourages sites to use these guidelines as early as possible in the application development process.

EU Data Protection

The European Commission's Directive on Data Protection protects the fundamental rights of European Union citizens to privacy with respect to the processing of personal data. The primary focus of the directive is on the acceptable use and protection of personal data. Like all other European Union privacy legislation, this directive also requires that personal data be collected, stored, changed or disseminated only with a citizen's express consent and with full disclosure as to the use of the data. The directive also prohibits the transfer of personal data from European organizations to non-European Union nations and organizations that do not adequately protect the safety and privacy of personal data. The United States has developed a Safe Harbor framework for U.S. organizations that are required to comply with this directive.

EU Directive on Privacy and Electronic Communications

European Union Directive on Privacy and Electronic Communications is part of a broader “telecoms package” of legislation that governs the electronic communications sector in the European Union. The directive reinforces a basic European Union principle that all member states must ensure the confidentiality of communications made over public communications networks and the personal and private data inherent in those communications. The directive governs the physical communication networks as well as the personal data that is carried on it.

FISMA

The United States Congress passed the E-Government Act of 2002 in recognition of the importance of information security to the economic and national security interests of the United States. Title III of the act, entitled the Federal Information Security Management Act (FISMA), tasked the National Institute of Standards and Technology with developing standards and guidelines to be used by all U.S. federal government agencies in implementing adequate information security as part of their information systems, underpinned by three security objectives for information systems: confidentiality, integrity and availability. FISMA requires the head of each federal agency to provide information security protections commensurate with the risk and magnitude of the harm that may result from unauthorized access, use, disclosure, disruption, modification or destruction of its information and information systems. The protection should apply not only within the agency, but also within contractor or other organizations working on behalf of the agency.

GLBA

The Gramm-Leach-Bliley Act (GLBA) mandates that financial institutions must protect consumers' personal financial information. The main provision affecting Web application security in the financial industry is the GLBA Safeguards Rule.

HIPAA

The Health Insurance Portability and Accountability Act (HIPAA) mandates the privacy and security of personal health information from the various threats and vulnerabilities associated with information management.

ISO17799

This is the most commonly accepted international standard for information security management. Use this policy as a baseline in crafting a compliance policy to meet the needs of your organization and its security policy.

ISO27001

ISO/IEC 27001 is an information security management system standard published in October 2005 by the International Organization for Standardization and the International Electrotechnical Commission. The basic objective is to help establish and maintain an effective information management system using a continual improvement approach. ISO 27001 specifies the requirements for the security management system itself. It is the standard, as opposed to ISO 17799, against which certification is offered. Additionally, ISO 27001 is "harmonized" with other management standards, such as ISO 9001 and ISO 14001.

JPIPA

Japan enacted the Personal Information Protection Act (JPIPA) in 2003 to protect individuals' rights and personal information while preserving the usefulness of information technology and personal information for legitimate purposes. The law establishes responsibilities for businesses that handle personal information for citizens of Japan and outlines potential fines and punishments for organizations that do not comply. The act requires businesses to communicate their purpose in collecting and using personal information. They must also take reasonable steps to protect personal information from disclosure, unauthorized use or destruction.

NERC

The North American Electric Reliability Council (NERC) was established in 1968 with the mission of ensuring that the electric system of the United States is reliable, adequate and secure. After President Bill Clinton issued Presidential Decision Directive 63 in 1998 to define infrastructure industries critical to the United States' national economy and public well-being, the U.S. Department of Energy designated the NERC to act as the coordinating agency for the electricity industry, which was named one of the eight critical infrastructure industries.

NIST 800-53

The United States Congress passed the E-Government Act of 2002 in recognition of the importance of information security to the economic and national interests of the United States. Title III of the act, entitled the Federal Information Security Management Act (FISMA), tasked the National Institute of Standards and Technology with developing standards and guidelines to be used by all U.S. federal government agencies in implementing adequate information security as part of their information systems, underpinned by three security objectives for information systems: confidentiality, integrity, and availability.

OMB

This policy addresses major application security sections that were defined in December 2004 by the Office of Management and Budget for federal agency public Web sites. These are information resources funded in whole or in part by the federal government and operated by an agency, contractor, or other organization on behalf of the agency. They present government information or provide services to the public or a specific non-federal user group and support the proper performance of an agency function.

OWASP Top Ten 2004/2007/2010

Many government agencies suggest testing for the Open Web Application Security Project (OWASP) Top Ten Web application vulnerabilities as a best practice in ensuring the security of your Web application.

PCI Data Security 1.2, 2.0

The Payment Card Industry (PCI) Data Security Policy requires that all PCI Data Security members, merchants, and service providers that store, process or transmit cardholder data verify all purchased and custom Web applications, including internal and external applications.

PIPEDA

Canada's Personal Information Protection and Electronic Documents Act (PIPEDA) is a new law that protects personal information in the hands of private sector organizations and provides guidelines for the collection, use and disclosure of that information in the course of commercial activity. The Act, based on ten privacy principles developed by the Canadian Standards Association, is overseen by the Privacy Commissioner of Canada and the Federal Court. As of January 1, 2004, all Canadian businesses are required to comply with the privacy principles set out by PIPEDA. The Act covers both traditional, paper-based and on-line business.

Safe Harbor

The European Commission's Directive on Data Protection prohibits the transfer of personal data from European organizations to non-European Union nations and organizations that do not adequately protect the safety and privacy of personal data. Upon passage of this comprehensive European legislation, all businesses and organizations in the United States that share data with European Union organizations were obligated to comply with the regulations, which could have disrupted many types of trans-Atlantic business transactions. Due to the differences in approaches taken by the United States and European Union nations in protecting personal data privacy, the U.S. Department of Commerce, in consultation with the European Commission, developed a streamlined "Safe Harbor" framework through which U.S. organizations could comply with the Directive on Data Protection.

Organizations participating in the Safe Harbor are committed to complying with these seven principles designed to ensure that personal data is properly used, controlled and protected: Notice, Choice, Onward Transfer, Access, Security, Data Integrity and Enforcement. Of particular significance to information technology:

- The Notice principle requires organizations to inform individuals about the purposes for which it collects information, such as through a privacy policy.
- The Security principle states that organizations will take reasonable precautions to protect personal data.
- The Enforcement principle mandates that organizations have procedures in place for verifying that security commitments are satisfied, such as through comprehensive security testing.

SANS CWE Top 25

The SANS (SysAdmin, Audit, Network, Security) Institute was established in 1989 as a cooperative research and education organization. The SANS Common Weakness Enumeration (CWE) Top 25 Most Dangerous Software Errors is a list of the most widespread and critical programming errors that can lead to serious software vulnerabilities. They are dangerous because they frequently allow attackers to completely take over the software, steal data, or prevent the software from functioning. This compliance template reports all applicable web application components of this list.

Sarbanes-Oxley

The Sarbanes-Oxley Act, which falls under the umbrella of the U.S. Securities and Exchange Commission (SEC), was enacted on July 30, 2002. It focuses on regulating corporate behavior for the protection of financial records, rather than enhancing the privacy and security of confidential customer information. For more information on using HP scanners to achieve Sarbanes-Oxley compliance for your Web applications, read the Sarbanes-Oxley white paper.

UK Data Protection

The European Commission's Directive on Data Protection protects the fundamental rights of European Union citizens to privacy with respect to the processing of personal data. The primary focus of the directive is on the acceptable use and protection of personal data. The United Kingdom implemented the protections mandated by the directive through its Data Protection Act of 1998, summarized as follows:

- Personal data should be processed fairly and lawfully and only with consent.

- Personal data should be obtained only for specified and lawful purposes, and should not be further processed in any manner incompatible with those purposes.
- Personal data should be adequate, relevant and not excessive in relation to the purpose or purposes for which they are processed.
- Personal data should be accurate and kept up to date.
- Personal data processed for any purpose should not be kept for longer than is necessary for that purpose.
- Personal data should be processed in accordance with the rights of data subjects.
- Appropriate technical and organizational measures should be taken against unauthorized or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.
- Personal data should not be transferred to a country or territory outside the European Economic Area unless that country or territory ensures an adequate level of protection for the rights and freedoms of data subjects in relation to the processing of personal data.

[WASC](#)

This compliance template is based on the Web Application Security Consortium threat classes. The WASC Threat Classification is a cooperative effort to clarify and organize the threats to the security of a web site. When used in conjunction with the All Checks policy, you can generate a compliance report that includes each vulnerability check contained in SecureBase.

[License Management](#)

When you first install and start WebInspect, you are prompted to enter a license token sent to you by HP. After you enter the token and customer information, the application contacts HP servers and downloads license information pertaining to your specific installation.

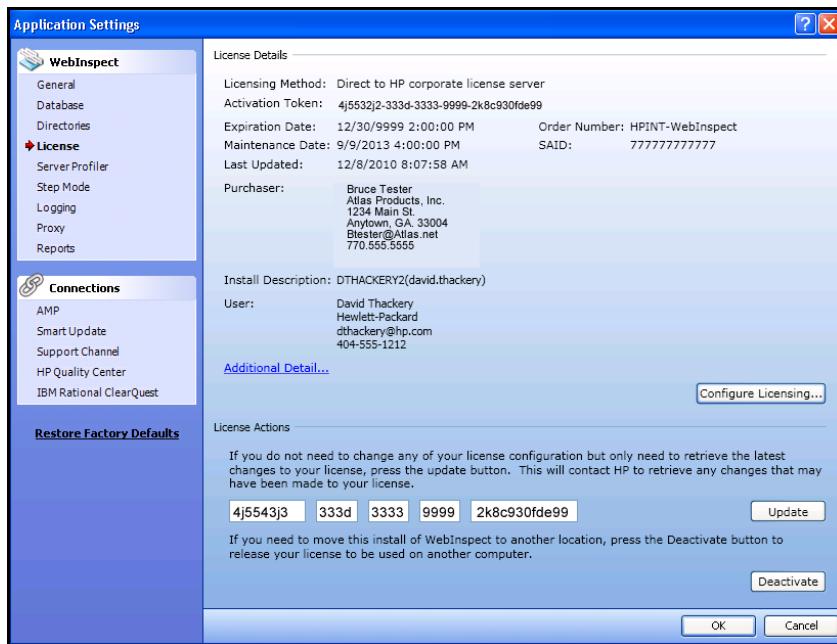
There are two methods for controlling WebInspect licenses:

- Connect directly to the HP corporate license server
- Connect to a local License and Infrastructure Manager.

To modify your license:

- 1 Click **Edit → Application Settings**.
- 2 Select the **License** category.

Connected to HP



Updating the License

If you upgrade from a trial version or if you otherwise modify the conditions of your license, you must click **Update** (a hyperlink near the bottom right corner) to update your license. The application will again contact HP and update the information stored locally on your machine.

Deactivating the License

WebInspect licenses are assigned to specific computers. If you would like to transfer this license to a different computer:

- 1 Copy the activation token.
 Take care not to lose or misplace this number. Write it or print it.
- 2 Click **Deactivate**.
- 3 At the new computer, access the WebInspect application settings for licensing and enter the activation token.

Configuring the License

- 1 Click **Configure Licensing**.
- 2 In the Licensing Method group, choose **Connect directly to HP corporate license server**.
- 3 Enter the information requested in the User Information group.
- 4 Click **Next**.
- 5 In the Activation Token area, enter the 32-digit license token sent to you by e-mail from HP. Omit any hyphens that may appear in the string (or simply copy the token, position your cursor in the first block of the Activation Token field, and press Ctrl + V).

- 6 If the machine you are attempting to license is connected to the Internet, select **Online Activation**.

 - a The default URL of the HP license service is `https://LicenseService.HPSmartUpdate.com`. Change this only if directed to do so by HP Support personnel.
 - b If you access the Internet through a proxy, select **Network Proxy** and choose a setting from the **Proxy Profile** list. You must also click **Edit** if you select **Use PAC file** or **Use Explicit Proxy Settings**.
- 7 If WebInspect is installed on a computer that is not connected to the Internet, select Offline Activation. You will create a license request file containing information about that computer and then, using a separate Internet-connected computer, access an HP Web site to transmit the file to an HP server, which will download a license file that you can copy and install on the computer that is not connected to the Internet.
 - a Click the **Browse** button next to the **License Request File** box.
 - b Select a location where the file will be saved. The name of the request file is formatted as `<ProductName>_LicenseReq.xml`.

Be sure to save this file on a portable media or at a location that is accessible by a machine that has access to the Internet.

For additional instructions, see Complete Offline License Activation.
- 8 Click **Next**.
- Information pertaining to your installed license appears in the License Details section.
- 9 Click **Finish**.

Complete Offline Activation

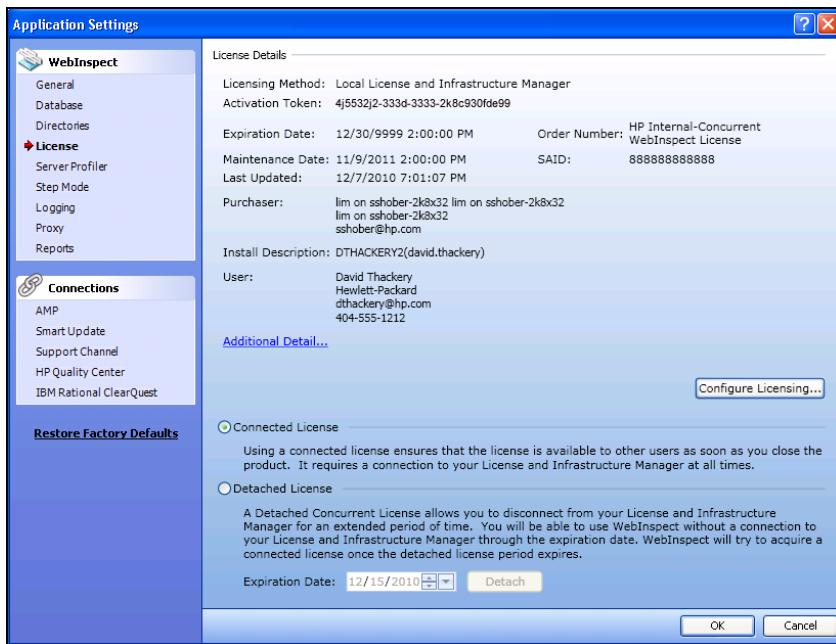
To download and install a license:

- 1 Go to a computer that is connected to the Internet.
- 2 Start Internet Explorer (or another browser) and navigate to `https://licenseservice.hpsmartupdate.com/OfflineLicensing.aspx`.

Make certain that this computer can access the request license file you created in the previous step.
- 3 Follow the instructions to create and download a response license file.

Make certain to download the file to an intranet location or portable media that is accessible by the computer you are attempting to license.
- 4 Return to the computer that is not connected to the Internet.
- 5 Under Install License, click the **Browse** button and select the response license file you just downloaded. The name of the file is formatted as `<ProductName>_LicenseResp.xml`.
- 6 Click **Next**.

Connected to License and Infrastructure Manager



Selecting Detached or Connected License

Select the manner in which you want the License and Infrastructure Manager to handle the license associated with the HP product installed on this computer.

- **Connected License** - The computer can run the HP product only when the computer is able to contact the LIM. Each time you start the HP software, the LIM allocates a seat from the license pool to this installation. When you close the software, the seat is released from the computer and allocated back to the pool, allowing another user to consume the license.
- **Detached License** - The computer can run the HP product anywhere, even when disconnected from your corporate intranet (on which the LIM is normally located), but only until the expiration date you specify. This allows you to take your laptop to a remote site and run the HP software. When you reconnect to the corporate intranet, you can access the Application License settings and reconfigure from Detached to Connected.

Configuring the License

- 1 Click **Configure License**.
- 2 In the Licensing Method group, choose **Connect to License and Infrastructure Manager**.
- 3 Enter the information requested in the User Information group.
- 4 Click **Next**.

The HP License and Infrastructure Manager (LIM) allows your company to manage concurrent licenses for HP software in a manner that best suits your organization's development and testing environment. Contact your LIM administrator to obtain the information required for completing this form.

- 5 In the **URL** box, enter the URL of the License and Infrastructure Manager.
- 6 Enter the name of the license pool and the pool password in the appropriate boxes.

- 7 If authorization is required to access the LIM, select **Network Credentials** and then enter your user name and password.
 - 8 If connecting to the LIM through a proxy, select **Network Proxy** and choose a setting from the **Proxy Profile** list. You must also click **Edit** if you select **Use PAC file** or **Use Explicit Proxy Settings**.
 - 9 Click **Next**.
 - 10 Select the manner in which you want the License and Infrastructure Manager to handle the license associated with the HP product installed on this computer.
 - **Connected License** - The computer can run the HP product only when the computer is able to contact the LIM. Each time you start the HP software, the LIM allocates a seat from the license pool to this installation. When you close the software, the seat is released from the computer and allocated back to the pool, allowing another user to consume the license.
 - **Detached License** - The computer can run the HP product anywhere, even when disconnected from your corporate intranet (on which the LIM is normally located), but only until the expiration date you specify. This allows you to take your laptop to a remote site and run the HP software. When you reconnect to the corporate intranet, you can access the Application License settings and reconfigure from Detached to Connected.
 - 11 Click **Next**.
- Information pertaining to your installed license appears in the License Details section.
- 12 Click **Finish**.

Command Line Execution

You can initiate several WebInspect functions via a command line interface using the program WI.exe. Use the following syntax when typing a command:

```
wi.exe -u url [-s file] [-ws file] [-Framework name] [-CrawlCoverage name][-ps policyID | -pc path] [-ab | an | am | ad | aa | ak {creds}] [-o | c][-n name] [-e[abcdefghijklmnol] file] [-x | xd | xa | xn] [-b filepath] [-v] [-?] [-r report_name -y report_type -w report_favorite -f report_export_file -g[phacxe][-t compliance_template_file]] [-d filepath -m filename][-i scanid] [-ir scanid] [-db]
```

To run multiple scans from the command line, create and execute a batch file, using a format similar to the following:

```
c:  
cd \program files\HP\HP Webinspect  
wi.exe -u http://172.16.60.19 -ps 4  
wi.exe -u http://www.mywebsite.com  
wi.exe -u http://172.16.60.17  
wi.exe -u http://172.16.60.16
```

The options are defined in the following table. Items in italics require a value.

Table 1: Command Line Syntax

Category	Parameter	Definition
General	?	Show usage.
	u {url}	URL (include host/port/scheme) or IP address. Caution 1: When using the -u parameter with -s (a settings file), be sure to specify an -x, -xa, -xd, or -xn parameter to restrict a scan to folders, if desired. Failure to do so may result in an unrestricted audit under certain conditions. Caution 2: If the URL contains an ampersand (&), you must enclose the URL within quotation marks.
	s {filename}	Settings file. Note: Command line parameters take precedence over values in a settings file.
	ws {filename}	Web Service Design file
	db	Use database defined in settings file. If omitted, WebInspect defaults to database connection defined in application settings.
	o	Audit only.
	c	Crawl only.
	n {name}	Scan name.
	b {filepath}	Use given SecureBase file.
	d {filepath}	Move database to filepath.
	m {filename}	Move database to filename.
	i (scanid)	Scan ID (GUID).
	ir (scanid)	Resume scan with the specified ID (GUID).
Restrict to Root Folder	x	Restrict to directory only (self).
	xa	Restrict to directory and parents (ancestors).
	xd	Restrict to directory and subdirectories (descendants).
	xn	Ignore “restrict to folder” rules in referenced settings file. Restrict to folder parameters (x xa xb xn) can be in their own category (as report or output).

Table 1: Command Line Syntax (cont'd)

Category	Parameter	Definition
Framework	framework {frameworkname}	Name of framework; currently only Oracle ADF Faces (Oracle) and IBM WebSphere Portal (WebSpherePortal) are supported. Optimizes scanning of application built with either of these technologies.
Crawl Coverage	CrawlCoverage {Coveragename}	Values for Coveragename are: Thorough = Exhaustive crawl of entire site. Default = Focus more on coverage than performance. Moderate = Balance of coverage and speed. Quick = Focus on breadth and performance.
Audit Policy	ps {id}	Use a non-customized policy. Possible values for <i>id</i> are: 1 = Standard 2 = Assault 3 = SOAP 4 = Quick 5 = Safe 6 = Development 7 = Blank 16 = QA 17 = Application 18 = Platform 1001 = SQL Injection 1002 = Cross-Site Scripting 1003 = OWASP Top 10 App. Security Risks 2007 1004 = All Checks 1005 = Passive 1008 = Critical and High Vulnerabilities 1009 = OWASP Top 10 App. Security Risks 2010 1010 = Aggressive SQL Injection 10000 = Hacme Bank
	pc {path}	Use a customized policy. For <i>path</i> , specify the full path and file name, such as: C:\MyPolicies\MyCustomPolicy.policy
Authentication	ab {id:pwd}	Basic mode (user name and password).
	an {id:pwd}	NTLM mode (user name and password).
	ad {id:pwd}	Digest mode (user name and password).
	aa {id:pwd}	Automatic mode (user name and password).
	ak {id:pwd}	Kerberos mode (user name and password).
	am {path}	Web macro authentication (path to macro).
Output	ea {filepath}	Export scan in full XML format.
	eb {filepath}	Export scan details (Full) in XML.

Table 1: Command Line Syntax (cont'd)

Category	Parameter	Definition
	ec {filepath}	Export scan details (Comments) in XML.
	ed {filepath}	Export scan details (Hidden Fields) in XML.
	ee {filepath}	Export scan details (Script) in XML.
	ef {filepath}	Export scan details (Set Cookies) in XML.
	eg {filepath}	Export scan details (Web Forms) in XML.
	eh {filepath}	Export scan details (URLs) in XML.
	ei {filepath}	Export scan details (Requests) in XML.
	ej {filepath}	Export scan details (Sessions) in XML.
	ek {filepath}	Export scan details (E-mails) in XML.
	el {filepath}	Export scan details (Parameters) in XML.
	em {folderpath}	Export scan details (Web Dump) in XML.
	en {folderpath}	Export scan details (Offsite Links) in XML.
	eo {folderpath}	Export scan details (Vulnerabilities) in XML.
	v	Verbose output.
Reports	r {name} For multiple reports, separate report names with a semicolon. All reports will be contained in a single file.	Values for <i>name</i> : Aggregate Alert View Attack Status Compliance Crawled URLs Developer Reference Duplicates Executive Summary False Positive QA Summary Scan Difference Scan Log Trend Vulnerability Vulnerability (Legacy) Note: Report names containing a space must be enclosed in quotation marks.
	w {name}	Name of the report favorite to run.
	ag	Aggregate reports in report favorite
	y {report_type}	The type of report (either “Standard” or “Custom”).
	f {export_path}	Where to save report file; full path and file name.
	gp	Export as pdf.

Table 1: Command Line Syntax (cont'd)

Category	Parameter	Definition
	gh	Export as HTML file (in ZIP file).
	ga	Export as raw report file.
	gc	Export as rich text file.
	gx	Export as text.
	ge	Export as Microsoft Excel file.
	t {filepath}	Use specified compliance template file.

If you do not specify a policy, WebInspect will crawl (but not audit) the Web site.

If you specify the name of a policy that does not exist, WebInspect will not conduct the scan.

To initiate a command from the command line:

- 1 Select **Run** from the Windows **Start** menu.
- 2 Type **cmd** in the **Open** box.
- 3 Click **OK**.
- 4 Navigate to the directory in which WebInspect is installed. The default directory is C:\Program Files\HP\HP WebInspect.

This will ensure proper handling of long file names. The process, as it appears in the Task Manager, is WI.exe. Scan data will be cached temporarily in the Working directory and then moved to the Scans directory.

Hyphens in Command Line Arguments

You can use hyphens in command line arguments (output files, etc.) only if the argument is enclosed in double quotes, as illustrated by the “export path” argument in the following command:

```
wi.exe -u http://zero.webappsecurity.com -ea "c:\temp\command-line-test-export.xml"
```

Uninstalling WebInspect

When uninstalling, you can choose to repair WebInspect or remove it from your computer.

If you select **Remove**, you may choose one or both of the following options:

- **Remove product completely** - Deletes the WebInspect application and all related files, including scan data stored on a local (non-shared) SQL server, settings files and logs.
- **Deactivate license** - Releases your WebInspect license, which allows you to install WebInspect on a different computer. Application data and files are not deleted.

4 Scanning a Site

Introduction

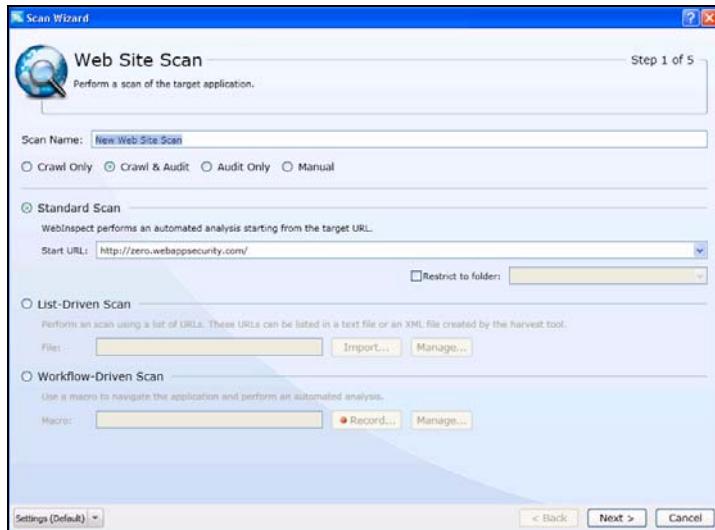
After updating your database and preparing for the scan, you are ready to determine your Web application's security vulnerabilities. Follow the steps below to start a scan:

- 1 On the WebInspect menu bar, click **File** → **New**.
- 2 Select the type of scan you want to conduct. The choices are:
 - **Web Site Scan**
 - **Web Service Scan**
 - **Enterprise Scan**

➤ Alternatively, you can click the **New** drop-down arrow on the toolbar and select a scan type, or select the appropriate command on the **Start** page.

Web Site Scan

When you start a Web Site scan, the *Scan Wizard* window appears.



The options displayed by default on this and subsequent windows are extracted from the WebInspect default settings. Any changes you make will be used for this scan only. If you click **Settings (Default)** at the bottom of the window to access the WebInspect *Settings* window, any selections you make are temporary. To change the default settings, you must select **Default Scan Settings** from the **Edit** menu.

Task 1: Choose Scan Options

- 1 In the **Scan Name** box, enter a name or a brief description of the scan.
- 2 (Optional) To load a settings configuration that you previously saved:
 - a Click the drop-down arrow on the **Settings** button.
 - b Do one of the following:
 - Select **From File** to open the standard file-selection window, then choose a file and click **Open**.
 - Select a file from the drop-down list.
- 3 Select one of the following scan modes:
 - **Crawl Only:** This option completely maps a site's hierarchical data structure. After a crawl has been completed, you can click **Audit** to assess an application's vulnerabilities.
 - **Crawl and Audit:** WebInspect maps the site's hierarchical data structure and audits each resource (page). Depending on the default settings you select, the audit can be conducted as each resource is discovered or after the entire site is crawled. For information regarding simultaneous vs. sequential crawl and audit, see [Crawl and Audit Mode](#) on page 166.
 - **Audit Only:** WebInspect applies the methodologies of the selected policy to determine vulnerability risks, but does not crawl the Web site. No links on the site are followed or assessed.
 - **Manual:** Manual mode allows you to navigate manually to whatever sections of your application you choose to visit, using Internet Explorer. It does not crawl the entire site, but records information only about those resources that you encounter while manually navigating the site. This feature is used most often to enter a site through a Web form logon page or to define a discrete subset or portion of the application that you want to investigate. Once you finish navigating through the site, you can audit the results to assess the security vulnerabilities related to that portion of the site that you recorded. See [Manual Scan](#) on page 155 for additional information.
- 4 Select one of the following scan types:
 - **Standard Scan:** WebInspect performs an automated analysis, starting from the target URL. This is the normal way to start a scan.
 - **Manual Scan:** Manual Crawl (Step Mode) allows you to navigate manually to whatever sections of your application you choose to visit, using Internet Explorer. This choice appears only if you select the Manual Scan mode (above).
 - **List-Driven Scan:** Perform a scan using a list of URLs to be scanned. Each URL must be fully qualified and must include the protocol (for example, http:// or https://). You can use a text file, formatted as comma-separated list or one URL per line, or the XML file generated by the FilesToURLs utility.
 - **Workflow-Driven Scan:** WebInspect audits only those URLs included in the \$\$macro\$\$ that you previously recorded and does not follow any hyperlinks encountered during the audit. A logout signature is not required. This type of macro is used most often to focus on a particular subsection of the application. You can select multiple macros.
- 5 If you select **Standard Scan**, follow these instructions:
 - a In the **Start URL** box, type or select the complete URL or IP address of the site you want to examine.

If you enter a URL, it must be precise. For example, if you enter MYCOMPANY.COM, WebInspect will not scan WWW.MYCOMPANY.COM or any other variation (unless you specify alternatives in the Allowed Hosts setting).

An invalid URL or IP address will result in an error. If you want to scan from a certain point in your hierarchical tree, append a starting point for the scan, such as <http://www.myserver.com/myapplication/>.

Scans by IP address will not pursue links that use fully qualified URLs (as opposed to relative paths).

WebInspect supports both Internet Protocol version 4 (IPV4) and Internet Protocol version 6 (IPV6). IPV6 addresses must be enclosed in brackets. See [Internet Protocol Version 6](#) on page 164.

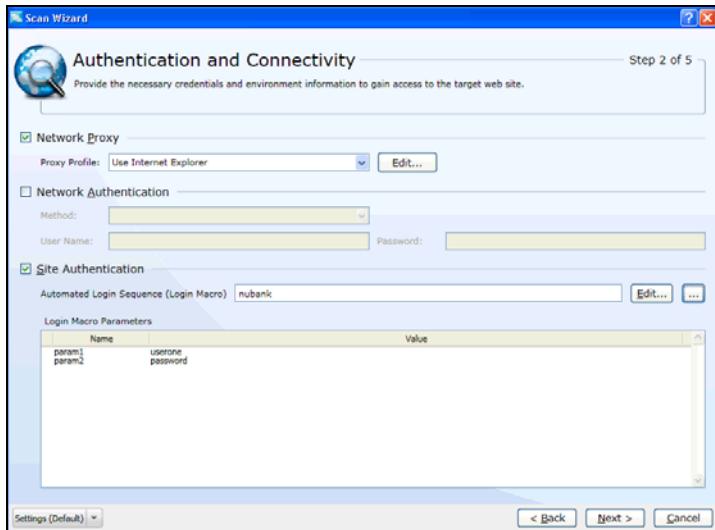
- b** If you select **Restrict to folder**, you can limit the scope of the scan to the area you choose from the drop-down list. The choices are:

 - **Directory only**—WebInspect will crawl and/or audit only the URL you specify. For example, if you select this option and specify a URL of www.mycompany/one/two/, WebInspect will assess only the “two” directory.
 - **Directory and subdirectories**—WebInspect will begin crawling and/or auditing at the URL you specify, but will not access any directory that is higher in the directory tree.
 - **Directory and parent directories**—WebInspect will begin crawling and/or auditing at the URL you specify, but will not access any directory that is lower in the directory tree.
- 6** If you select **Manual Scan**, enter a URL and, if desired, select **Restrict to folder**. See Standard Scan (above).
- 7** If you select **List-Driven Scan**, do one of the following:

 - Click **Import** and select a text file or XML file containing the list of URLs you want to scan.
 - Click **Manage** to create or modify a list of URLs.
- Note: This feature is also used in conjunction with the FilesToURLs utility. See [FilesToURLs Utility](#) on page 162 for more information.
- 8** If you select **Workflow-Driven Scan**, do one of the following:

 - Click **Manage** to select, edit, record, import, export, or remove a macro.
 - Click **Record** and create a macro.
- Note: You can include more than one macro in a scan.
- 9** Click **Next**.

Task 2: Provide Authentication and Connectivity Information



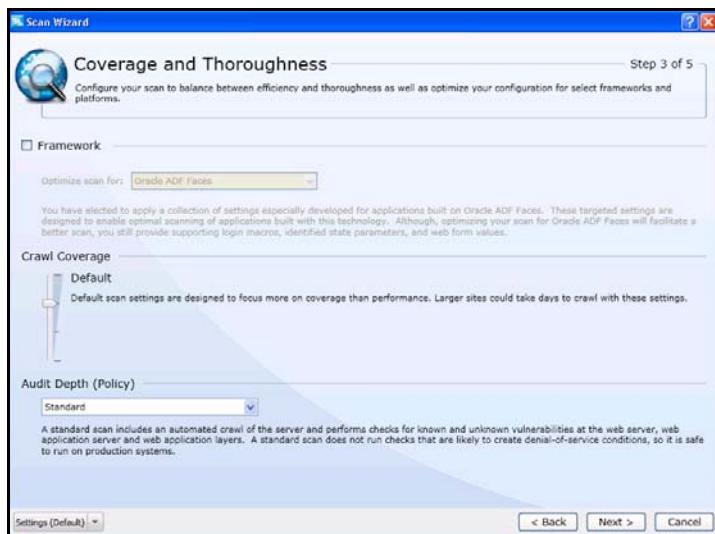
- 1 If you need to access the target site through a proxy server, select **Network Proxy** and then choose an option from the **Proxy Profile** list:
 - **Autodetect**—Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's Web proxy settings.
 - **Use Internet Explorer**—Import your proxy server information from Internet Explorer.
 - **Use PAC File**—Load proxy settings from a Proxy Automatic Configuration (PAC) file. If you select this option, click **Edit** to enter the location (URL) of the PAC.
 - **Use Explicit Proxy Settings**—Specify proxy server settings. If you select this option, click **Edit** to enter proxy information.
 - **Use Mozilla Firefox**—Import your proxy server information from Firefox.
- 2 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," or if the Internet Explorer setting "Use a proxy server for your LAN" is not selected, then a proxy server will not be used.
- 3 Select **Network Authentication** if server authentication is required. Then select an authentication method and enter your network credentials.
- 4 Select **Site Authentication** to use a recorded macro containing a user name and password that allows you to log on to the target site. The macro must also contain a "logout condition," which indicates when an inadvertent logout has occurred so WebInspect can rerun this macro to log on again.
 - Click **[...]** to select a macro.
If, after selecting a macro, you want to modify the macro using the Web Macro Recorder, click **Edit**.
 - Click **Record...** to create a macro.
 - The **Login Macro Parameters** grid appears if, when recording the macro, you selected the Smart Credentials option (when using the traffic-mode or event-based web macro recorders) or if you created input parameters when using the TruClient web macro recorder. Enter a user name and password. When scanning the page containing the

input control associated with this entry, WebInspect will substitute these credentials for those used in the macro. This feature allows you to create a macro using your user name and password, yet when other persons run the scan using this macro, they can substitute their own user credentials.

Note: For help creating login parameters with the TruClient web macro recorder, see [Using Name and Password Parameters](#) on page 344. For help with Smart Credentials using the event-based web macro recorder, see [Macro Settings](#) on page 340. For help with Smart Credentials using the traffic-mode web macro recorder, see Step 2 of “[Inspecting and Editing a Macro](#)” on page 325.

- 4 Click **Next**.

Task 3: Specify Coverage and Thoroughness



- 1 To optimize settings for an application built using either Oracle Application Development Framework Faces components or IBM WebSphere Portal, select **Framework** and then choose **Oracle ADF Faces** or **WebSphere Portal** from the **Optimize scan for** list. HP may develop other settings overlays and make them available through Smart Update.
- 2 Use the **Crawl Coverage** slider to specify the crawler settings.

This slider may or may not be enabled, depending on the scan mode you selected. The label associated with this slider also depends on your selection. If enabled, the slider allows you to select one of four crawl positions. Each position represents a specific collection of settings, as represented by the following labels:

Thorough

Thorough uses the following settings:

- Redundant Page Detection: OFF
- Maximum Single URL Hits: 20
- Maximum Web Form Submissions: 7
- Create Script Event Sessions: ON
- Maximum Script Events Per Page: 2000

- Number of Dynamic Forms Allowed Per Session: Unlimited
Note: A dynamic form is an HTML form that is modified when executing scripts. If it seems to take a long time to audit a script-intensive site, try using a setting that limits the number of dynamic forms per session to zero or one.
- Include Parameters In Hit Count: True

Default

Default uses the following settings:

- Redundant Page Detection: OFF
- Maximum Single URL Hits: 5
- Maximum Web Form Submissions: 3
- Create Script Event Sessions: OFF
- Maximum Script Events Per Page: 1000
- Number of Dynamic Forms Allowed Per Session: Unlimited
- Include Parameters In Hit Count: True

Normal

Normal uses the following settings:

- Redundant Page Detection: OFF
- Maximum Single URL Hits: 5
- Maximum Web Form Submissions: 2
- Create Script Event Sessions: OFF
- Maximum Script Events Per Page: 300
- Number of Dynamic Forms Allowed Per Session: 1
- Include Parameters In Hit Count: False

Quick

Quick uses the following settings:

- Redundant Page Detection: ON
- Maximum Single URL Hits: 3
- Maximum Web Form Submissions: 1
- Create Script Event Sessions: OFF
- Maximum Script Events Per Page: 100
- Number of Dynamic Forms Allowed Per Session: 0
- Include Parameters In Hit Count: False

If you click **Settings** (to open the *Advanced Settings* window) and change a setting that conflicts with any setting established by one of the four slider positions, the slider creates a fifth position labeled **Customized Coverage Settings**.

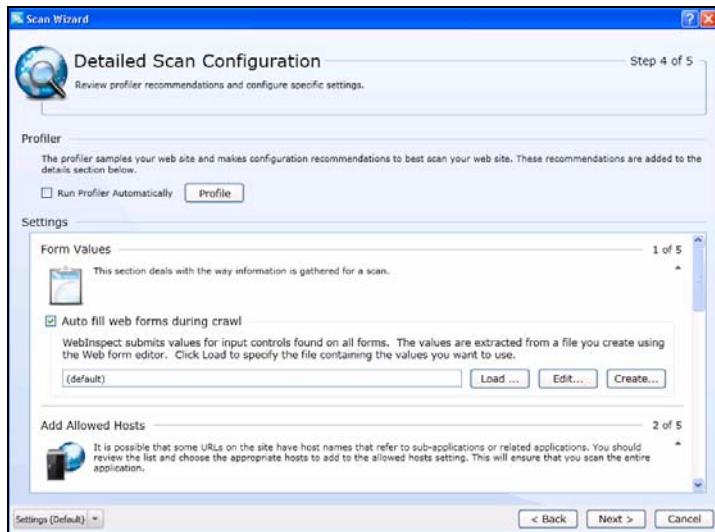
- 3 Select a policy from the **Audit Depth (Policy)** list.

This list may or may not be enabled, depending on the scan mode you selected in Step 1.

For a description of policies, see [Appendix B, Policies and Components](#).

- 4 Click **Next**.

Task 4: Provide Detailed Scan Configuration



If you click **Profile** (or if **Run Profiler Automatically** is selected when this page appears), WebInspect conducts a preliminary examination of the target Web site to determine if certain 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 suggestion to configure the required information before continuing.

Similarly, your settings may specify that 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 WebInspect setting to accommodate this feature.

To launch the Profiler each time you access this page, select **Run Profiler Automatically**.

To launch the Profiler manually, click **Profile**. See [Server Profiler](#) on page 363 for detailed information.

Results appear in the **Settings** section.

- 1 Accept or reject the suggestions. To reject, clear the associated check box.
- 2 If necessary, provide the requested information.
- 3 Click **Next**.

Several options may be presented even if you do not run the Profiler, depending on the settings you selected. They include:

- Auto fill Web forms
- Add allowed hosts
- Reuse Identified false positives
- Apply sample macro
- Traffic analysis

Auto Fill Web Forms

Select Auto-fill Web forms during crawl if you want WebInspect to submit values for input controls on forms it encounters while scanning the target site. WebInspect will extract the values from a prepackaged default file or from a file that you create using the Web Form Editor. You may:

- Click  to locate and load a file.
- Click **Edit** to edit the selected file (or the default values) using the Web Form Editor.
- Click **Create** to open the Web Form Editor and create a file.

Add Allowed Hosts

Use the Allowed Host settings to add domains to be crawled and audited. If your Web presence uses multiple domains, add those domains here. See [Allowed Hosts](#) on page 181 for more information.

To add allowed domains:

- 1 Click **Add**.
- 2 On the *Specify Allowed Host* window, enter a URL (or a regular expression representing a URL) and click **OK**.

Reuse Identified False Positives

Select scans containing vulnerabilities that were changed to false positives. If those false positives match vulnerabilities detected in this scan, the vulnerabilities will be changed to false positives.

To reuse identified false positives:

- 1 Select **Import False Positives**.
- 2 Click **Select Scans**.
- 3 Select one or more scans containing false positives from the same site you are now scanning.
- 4 Click **OK**.

You cannot import false positives when scheduling a scan or conducting an Enterprise scan.

Sample Macro

WebInspect's example banking application, zero.webappsecurity.com, uses a Web form login. If you scan this site, select **Apply sample macro** to run the prepackaged macro containing the login script.

Traffic Analysis

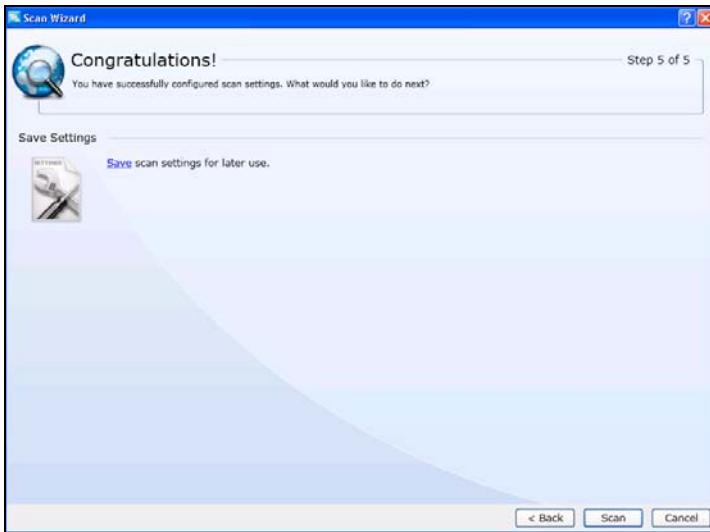
Select **Launch and Direct Traffic through Web Proxy** to use the Web Proxy tool to examine the HTTP requests issued by WebInspect and the responses returned by the target server.

While scanning, WebInspect displays in the navigation pane only those sessions that reveal the hierarchical structure of the Web site, plus those sessions in which a vulnerability was discovered. However, if you select **Enable Traffic Monitor**, WebInspect adds the **Traffic Monitor** button to the Scan Info panel, allowing you to display and review every single HTTP request sent by WebInspect and the associated HTTP response received from the server.

Message

If the profiler does not recommend changes, the Scan Wizard displays the message, “No settings changes are recommended. Your current scan settings are optimal for this site.”

Task 5: Initiate or Schedule the Scan



The contents of this window vary, depending your choices and configuration.

Upload to AMP/WebInspect Enterprise Scan Template

(Feature not illustrated) When connected to an enterprise server (AMP or to WebInspect Enterprise), you can send the settings for this scan to the enterprise server, which will create a scan template. However, you must be assigned to a role (in AMP or WebInspect Enterprise) that allows you to create scan templates. Click the **Upload** hypertext link.

Save Settings

You can save the settings you configured for this scan, which would allow you to reuse the settings for a future scan. Click the **Save** hypertext link.

Generate Reports

(Feature not illustrated) If you are scheduling a scan, you can instruct WebInspect to generate a report when the scan completes.

- 1 Select **Generate Reports**.
- 2 Click the **Select reports** hyperlink.
- 3 (Optional) Select a report from the **Favorites** list.

A “favorite” is simply a named collection of one or more reports and their associated parameters. To create a favorite once you have selected reports and parameters, click the **Favorites** list and select **Add to favorites**.
- 4 Select one or more reports.
- 5 Provide information for any parameters that may be requested. Required parameters are outlined in red.
- 6 Click **Next**.

- 7 If you select **Automatically Generate Filename**, the name of the report file will be formatted as <reportname> <date/time>. <extension>. For example, if creating a compliance report in pdf format and the report is generated at 6:30 on April 5, the file name would be “Compliance Report 04_05_2009 06_30.pdf.” This is useful for recurring scans.
- Reports are written to the directory specified for generated reports in the Application settings.
- 8 If you did not select **Automatically Generate Filename**, enter a name for the file in the **Filename** box.
- 9 Select the report format from the **Export Format** list.
- 10 If you selected multiple reports, you can combine them all into one report by selecting **Aggregate reports into one report**.
- 11 Select a template that defines the headers and footers used for the report and, if necessary, provide the requested parameters.
- 12 Click **Finished**.
- 13 Click **Schedule**.

WebSphere Portal Frequently Asked Questions

[How do you know if an application is running on WebSphere Portal?](#)

WebSphere Portal applications typically have very long urls that begin with /wps/portal or /wps/myportal followed by encoded sections. For example:

```
http://myhost.com/wps/portal/internet/customers/home!/ut/p/b1/
fY7BcoIwFAC_xS94T4QCx6Rpk6qlo20x5tIJShEIJoID0q-vnfFq97Yze1hQIEEddV8W-lzaozZ_
rh6-HjkRfrhERBZ4-EKESBmde5ggzEEVxmbXNGW7-sIsKdgTW3c_B3xmpzBfnacLv6QuIfx
VHKJGhmNfzToue8nWdKg4fx8jtaT9MJpB2zQPgqLp9GrADyey0tvvL1F9Snftm_y0cbuw8Xb
mvg2NN6412wlsQP27GAa3AO9AEBJhmxxcnWHlk8kverBIBQ!!/dl4/d5/
L2dBISEvZ0FBIS9nQSEh/
```

[Which versions of WebSphere Portal are supported?](#)

Versions 6.1 and later are supported.

[Why does WebInspect require special settings to scan a WebSphere Portal application?](#)

The encoded sections of the URL include what is called “navigation state,” which contains information about how to display elements in the current page (similar to VIEWSTATE in .Net) plus the navigation history. It is this navigation history that is troublesome for automated crawlers. As the crawler visits each link, the navigation state is being updated. This causes links on a page that the crawler may have already visited to continuously change. Since these look like new links, the crawler visits them and becomes trapped in an endless cycle.

When the WebSphere Portal overlay is selected, WebInspect can decode the navigation state in a URL and determine if the URL has already been visited. This prevents the crawler from continuously visiting the same page over and over again.

How does WebInspect decode the navigation state?

WebSphere Portal 6.1 and later include a URL decoding service. When the WebSphere Portal overlay is selected, WebInspect can pass a URL to the decoding service and evaluate the response to determine if this URL has already been visited. Although the decoding service is on by default, it is possible to turn it off in your WebSphere Portal server configuration. To get a good scan of your site with WebInspect, the decoding service must be enabled.

Is the navigation state just a special kind of session ID?

No. Navigation state does not contain any session information. Session is maintained via cookies.

Any special instructions when recording a login macro?

Make sure that the cookies JSESSIONID and LtpaToken are set as state parameters.

Why does the site tree contain deeply nested folders?

WebInspect's site tree does not currently understand how to parse the navigation state in WebSphere Portal URLs. It treats each section as a directory. These are, of course, not real directories. You will generally need to drill down to the lowest level of each branch to see the real content.

Is there any limitation on what types of attacks WebInspect can perform on WebSphere Portal applications?

WebInspect can perform all manipulation attacks on WebSphere Portal applications. This includes (but is not limited to) XSS, SQL Injection, CSRF, RFI, LFI and others. WebInspect will not perform any site search attacks when scanning a WebSphere Portal site. These include searching for backup files (.bak, .old), hidden files, hidden directories and platform specific configuration files. The reason for this exclusion is because almost any request will result in a 200 response to the default portal view and so there is no way to distinguish between an error response and a valid response.

How can you tell if the crawler is working correctly on a WebSphere Portal site?

The WebSphere Portal decoding service must be enabled and reachable on the server for the crawler to perform optimally. You can confirm if this is working by manually decoding a URL. Copy a URL from your site and modify it like this:

```
http://myhost.com/wps/poc?uri=state: path with navigation state>&mode=download
```

You should get an XML response. Alternatively, start a scan of your site with the WebSphere Portal overlay selected. Enable Traffic Monitor or run the scan through the Web Proxy. You should see periodic requests to the decoder service in the following format:

```
http://myhost.com/wps/poc?uri=state: path with navigation state>&mode=download.
```

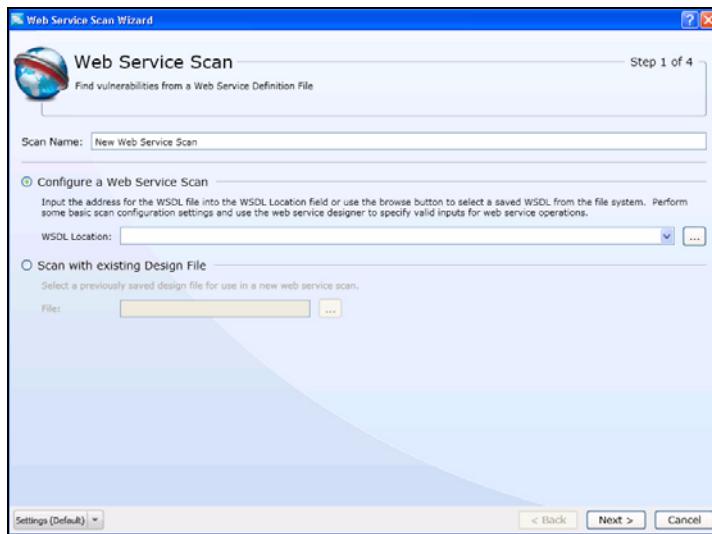
Another thing to consider is that the path of the decoding service can be changed on the server. If this is the case, you will need to modify your scan settings manually. Contact HP technical support for assistance.

It is also possible to modify the navigation state marker. By default this is !ut/p. If this is changed from the default on the server, you will need to modify your scan settings manually. Contact HP technical support for assistance.

Web Service Scan

When performing a Web service scan, WebInspect crawls the WSDL site and submits a value for each parameter in each operation it discovers. These values are extracted from a file that you must create using the Web Service Designer. It then audits the site by attacking each parameter in an attempt to detect vulnerabilities such as SQL injection.

When you start a Web service scan, the *Web Service Scan Wizard* window appears.

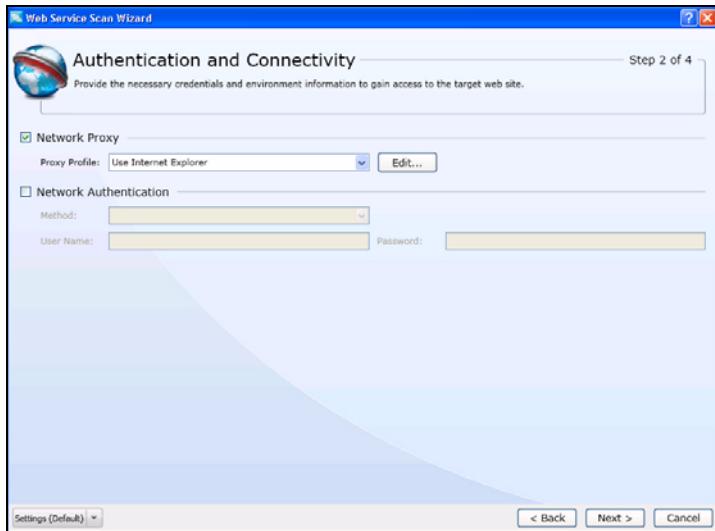


Task 1: Select a WSDL or Design File

- 1 Enter a name for this scan in the **Scan Name** box.
- 2 Select one of the following:
 - **Configure a Web Service Scan** - Enter or select the full path and name of a Web Service Definition Language (WSDL) file, or click [...] to open a standard file-selection dialog and choose a WSDL file. You will import the WSDL file and later launch the Web Service Test Designer to configure a file containing values for each operation in the service.
 - **Scan with Existing Design File** - Click [...] to open a standard file-selection dialog and choose a Web Service Test Design (WSD) file that you previously created using the Web Service Test Designer. This file contains values for each operation in the service.
- 3 Click **Next**.

Note: On any window presented by the Web Service Scan Wizard, you can click **Settings** (at the bottom of the window) to modify the default settings or to load a settings file that you previously saved. Any changes you make will apply to this scan only and will not be retained in the default settings file. To make and retain changes to default settings, click the WebInspect **Edit** menu and select **Default Scan Settings**.

Task 2: Provide Authentication and Connectivity Information

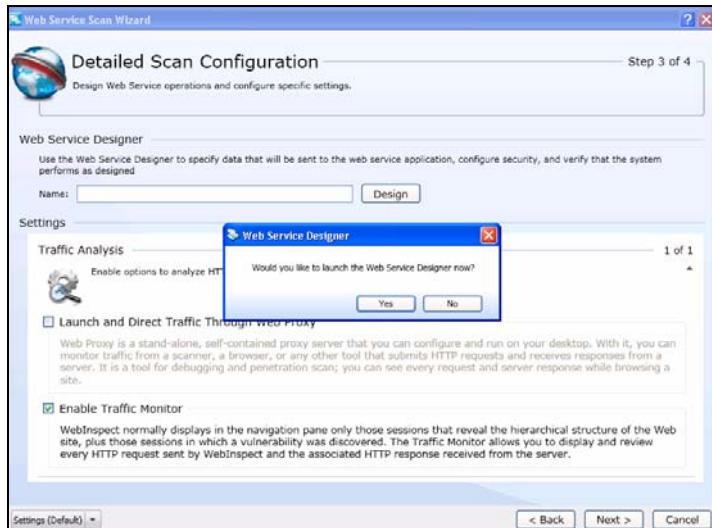


- 1 If you need to access the target site through a proxy server, select Network Proxy and then choose an option from the Proxy Profile list:
 - **Autodetect:** Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's Web proxy settings.
 - **Use Internet Explorer:** Import your proxy server information from Internet Explorer.
 - **Use PAC File:** Load proxy settings from a Proxy Automatic Configuration (PAC) file. If you select this option, click Edit to enter the location (URL) of the PAC.
 - **Use Explicit Proxy Settings:** Specify proxy server settings. If you select this option, click Edit to enter proxy information.
 - **Use Mozilla Firefox:** Import your 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," or if the Internet Explorer setting "Use a proxy server for your LAN" is not selected, then a proxy server will not be used.

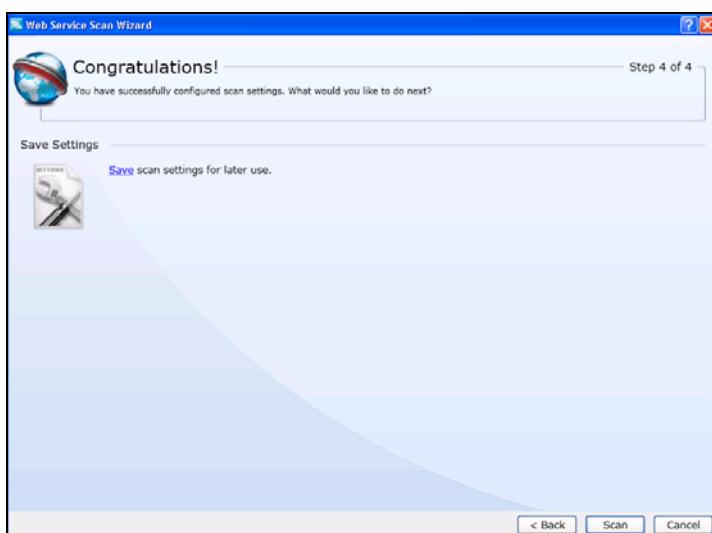
- 2 If server authentication is required, select Network Authentication and then select an authentication method and enter your network credentials.
- 3 Click **Next**.

Task 3: Detailed Scan Configuration



- 1 If you are creating a design test file, a message prompts you to launch the Web Service Test Designer. The Scan Wizard will not advance until you use the designer to create a web service design (WSD) file.
- 2 If you already selected a design test file, you may click **Design** to open the Web Service Test Designer and edit a WSD file containing values that should be submitted to the WSDL file during the scan.
- 3 (Optional) You may select the following options:
 - Launch and Direct Traffic through Web Proxy. Note: This option is not available if you are scheduling a scan.
 - Enable Traffic Monitor
- 4 Click **Next**.

Task 4: Initiate or Schedule the Scan



The contents of this window vary, depending your choices and configuration.

[Upload to AMP/WebInspect Enterprise Scan Template](#)

(Feature not illustrated) When connected to an enterprise server (AMP or WebInspect Enterprise), you can send the settings for this scan to the server, which will create a scan template. However, you must be assigned to a role (in AMP or WebInspect Enterprise) that allows you to create scan templates. Click the **Upload** hypertext link.

[Save Settings](#)

You can save the settings you configured for this scan, which would allow you to reuse the settings for a future scan. Click the **Save** hypertext link.

[Generate Reports](#)

(Feature not illustrated) If you are scheduling a scan, you can instruct WebInspect to generate a report when the scan completes.

To complete the procedure, click **Scan** (or click **Schedule**, if you are scheduling a scan).



WebInspect 9.30 offers minimal support for Web Service scans that were created with versions of WebInspect earlier than 9.0. These scans do not contain all the information required to render them properly in the current user interface and will exhibit the following attributes:

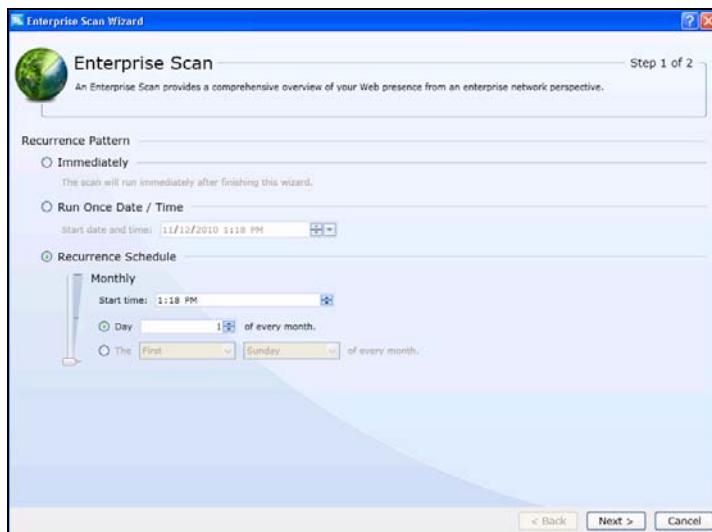
- The tree view may not show the correct structure.
- Even if the operations do not appear in the tree view, the vulnerabilities will appear in the vulnerability list. You should be able to select these vulnerabilities and view the vulnerability information, as well as the request and the response.
- Nothing will display in the XmlGrid.
- The rescan functionality should launch the Web Services scan wizard and select the first option having the selected WSDL already populated. This should force the Web Service Test Designer to open on page 3.
- The “Vulnerability Review” feature should be disabled.
- All reports should work as in previous WebInspect releases.
- The Scan view should render in “ReadOnly” mode, which disables the Start, Audit and Current Settings buttons.

If you conducted a Web Service scan using a version of WebInspect prior to 9.0, HP recommends that you rescan your Web service using WebInspect 9.30.

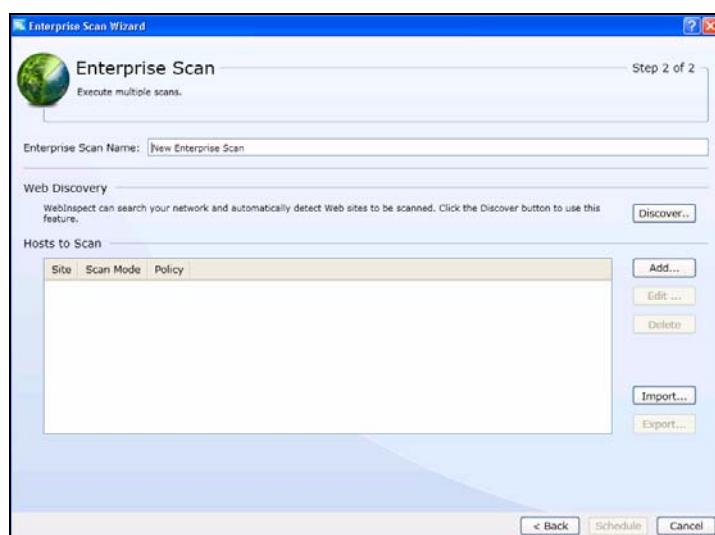
[Enterprise Scan](#)

An enterprise scan provides a comprehensive overview of your Web presence from an enterprise network perspective. WebInspect will automatically discover all available ports for a range of IP addresses. You can then select which servers to assess for vulnerabilities from all servers that are discovered.

When you start an enterprise scan, the *Enterprise Scan Wizard* window appears. The first of two screens prompts you to configure a recurrence pattern.



- 1 Specify when you want to conduct the scan. The choices are:
 - **Immediately**
 - **Run Once**—Modify the date and time when the scan should begin. You can click the drop-down arrow to reveal a calendar for selecting the date.
 - **Recurrence Schedule**—Use the slider to select a frequency (Daily, Weekly, or Monthly). Then specify the time when the scan should begin and (for Weekly or Monthly) provide other schedule information.
- 2 Click **Next**.



- 3 In the **Enterprise Scan Name** box, enter a unique name for this enterprise scan.

Enterprise scans provide a comprehensive overview of your Web presence from an enterprise network perspective. You can:

- Instruct WebInspect to discover all available servers within a range of IP addresses and ports that you specify.

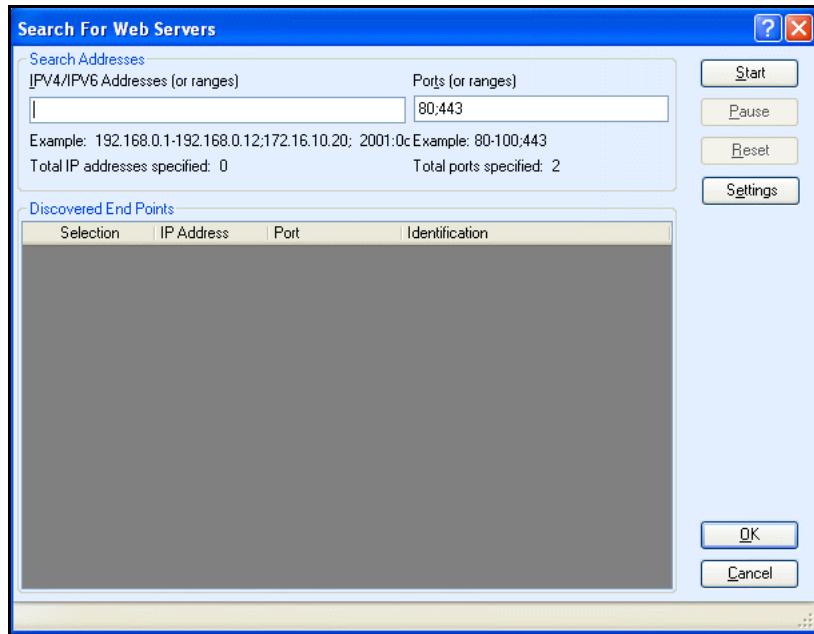
- Enter individual URLs or IP addresses.
- Import a list of servers (using a list that you previously created).

Web Discovery

Follow the steps below to discover Web servers.

- 1 Click **Discover**.

The *Search for Web Servers* window appears.



- 2 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.
- 3 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.
- 4 (Optional) Click **Settings** to modify the number of sockets and timeout parameters used for the discovery process.
- 5 Click **Start** to initiate the discovery process.

Results display in the Discovered End Points area.

- Click an entry in the **IP Address** column to view that site in a browser.
 - Click an entry in the **Identification** column to open the *Session Properties* window, where you can view the raw request and response.
- 6 To remove a server from the list, clear the associated check box in the **Selection** column.
- 7 Click **OK**.

Hosts to Scan

Follow the steps below to manually enter a list of URLs or IP addresses you want to scan.

- 1 Click **Add**.
- 2 Provide the information described in the Web Site Scan wizard.
- 3 Repeat for additional servers.

Import a List

If you previously used the Enterprise Scan feature or the Web Discovery tool to detect servers and then exported your findings to a text file, you can load those results by clicking **Import** and then selecting the saved file.

Edit the 'Hosts to Scan' List

After building a list of servers using one or more of the above methods, you can modify the list using the following procedure:

To modify the settings for a specific scan:

- 1 Select the server.
- 2 Click **Edit**.
- 3 Change the settings.
- 4 Click **Finish** (on the *Edit Web Site Scan* window)

To delete a server from the list:

- 1 Select a server.
- 2 Click **Delete**.

Export a List

To save the "Hosts to Scan" list:

- 1 Click **Export**.
- 2 Using a standard file-selection window, specify the file name and location.

Start the Scan

To begin the enterprise scan, click **Schedule**. Each server's scan results are saved automatically upon completion in your default Scans folder. The name of the server, along with a date and time stamp, is included in the file name.



WebInspect licenses permit users to scan specific IP addresses or a range of addresses. If a server has an IP address that is not permitted by your license, that server will not be included in the scan.

Manual Scan

A manual scan (also referred to as Step Mode) is a Web site scan option that allows you to navigate manually to whatever sections of your application you choose to visit. It does not crawl the entire site, but records information about only those resources that you encounter while manually navigating the site. This feature is used most often to enter a site through a login page or to define a discrete subset of the application that you want to investigate.

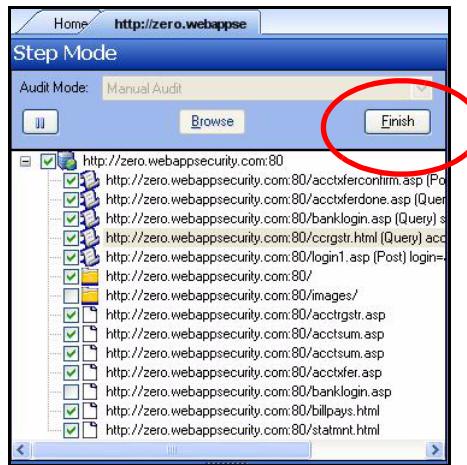
Once you finish navigating through the site, you can audit the results to assess the security vulnerabilities of the areas you visited and recorded.

Follow the steps below to record a site using step mode:

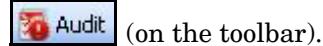
- 1 On the **WebInspect** menu bar, click **File** → **New** → **Web Site Scan**.

Alternatively, you can click the **WebInspect Start Page** tab and select **Start a Web Site Scan**.

- 2 Follow the instructions for configuring a Web site scan as described in [Web Site Scan](#) on page 137, selecting **Manual** as the scan method.
- 3 After configuring all scan parameters, click **Scan**.
- 4 When a browser opens, use it to navigate through the target site, visiting the areas you want to record.
- 5 When done, return to the *WebInspect* window and click **Finish** in the upper right corner of the navigation pane (Step Mode view).



- 6 When the navigation pane returns to the Site view, you can audit the site by clicking



(on the toolbar).

If you are using Step Mode to navigate a site that requires either Basic or NTLM authentication, WebInspect will remember the user name and password, and use them when you audit that site. Every server can have its own authentication scheme; WebInspect will distinguish between NTLM and Basic authentication on each server encountered during Step Mode.

Reviewing and Retesting Vulnerabilities

WebInspect offers several methods for reviewing or retesting discovered vulnerabilities. You may:

- Retest an individual vulnerability
- Retest all discovered vulnerabilities
- Rescan the entire site
- Compare two scans

Retest Individual Vulnerability

The retest feature is an extremely powerful tool for confirming that developers have fixed a specific vulnerability without having to conduct an entirely new scan.

- 1 Open a scan.
- 2 Right-click a vulnerable session in the Navigation pane or right-click a single vulnerability on the **Vulnerability** tab of the Summary pane.
- 3 Select **Review Vulnerability** from the shortcut menu.
- 4 On the *Vulnerability Review* window, click **Retest**.

WebInspect resubmits the entire vulnerability path to the server, compares each result to the original response, and displays the percentage of retest responses that match the original. This indicates whether the path to the vulnerability was accurately reproduced. Each HTTP request and response for the original session and the retest session can be compared side by side, instantly revealing any significant variations. Once the item has been confirmed as a vulnerability, you can submit the defect to either HP Quality Center (ALM) or IBM Rational ClearQuest.

For more information, see [Retesting/Reviewing Vulnerabilities](#) on page 85.

Retest All Vulnerabilities

This type of scan examines only those portions of the target site in which vulnerabilities were detected during the original scan. WebInspect does not conduct a new crawl of the site, but simply retraces the path of vulnerable sessions (as recorded in the original scan) and attacks the resources using the same checks previously employed.

Use the **Vulnerability** tab in the Summary pane to view the results. The grid contains an additional column named Reproducible, which may contain the following values:

- Not Found/Fixed - The vulnerability detected in the original scan was not found by the retest. These vulnerabilities are displayed with gray text. You can conduct a vulnerability review and retest of these items. The percentage in parentheses indicates a heuristic confidence level for this determination.
- Reproduced - Both the original scan and the retest detected the same vulnerability. In other words, the vulnerability still exists.
- New - The retest detected a vulnerability that was not reported in the original scan. This is most likely attributable to content that was added to the resource after the original scan was conducted.

Note: This bulk retest feature uses only those portions of a scan policy that revealed vulnerabilities in the original scan. If new vulnerabilities have been introduced since then, they may be detectable only by checks that were not used by the retest.

To retest all vulnerabilities:

- 1 Do one of the following:
 - Open a scan.
 - Select a scan on the Manage Scans page.
- 2 Click **Rescan** and select **Verify Vulnerabilities**.

The default name of the scan will be “Site Retest - <original scan name>”; for example, the retest of a site that originally resulted in a scan named MySite would produce a scan named Site Retest - MySite. However, you can specify a different name when launching the scan.

Rescan the Site

The Rescan feature allows you to transition easily from an open or selected scan into the scan wizard with the original scan settings preloaded. You may wish to conduct an identical scan of an updated site (using the same settings that were used for the original scan) to determine if previously discovered vulnerabilities have been fixed and if new ones have been introduced. Alternatively, you might want to tweak some of the settings to improve the crawl or audit.

The rescan functionality is available in two areas: the **Rescan** button on the scan toolbar and the **Rescan** button (and shortcut menu) for a selected scan on the Manage Scans pane.

- 1 Do one of the following:
 - Open a scan, click **Rescan** and select **Scan Again**.
 - On the WebInspect Start page, click **Manage Scans**; then select a scan and click **Rescan**.
- 2 Using the Scan Wizard, you may optionally modify the settings that were used for the original scan.

Note: The scan name is set by default to <original_scan_name>-1. If you conduct a rescan of a rescan, the integer appended to the default name will be incremented by one.

- 3 On the last step of the Scan Wizard, click **Scan**.

Note: You cannot rescan the results of a “Verify Vulnerabilities” function.

Compare Scans

You can compare the vulnerabilities revealed by two different scans of the same target and use this information to:

- Verify fixes - Compare vulnerabilities detected in the initial scan with those in a subsequent scan of a site in which the vulnerabilities were supposedly fixed.
- Check on scan health - Change scan settings and verify that those changes expand the attack surface.
- Find new vulnerabilities - Determine if new vulnerabilities have been introduced in an updated version of the site.
- Investigate Issues - Pursue anomalies such as false positives or missed vulnerabilities.

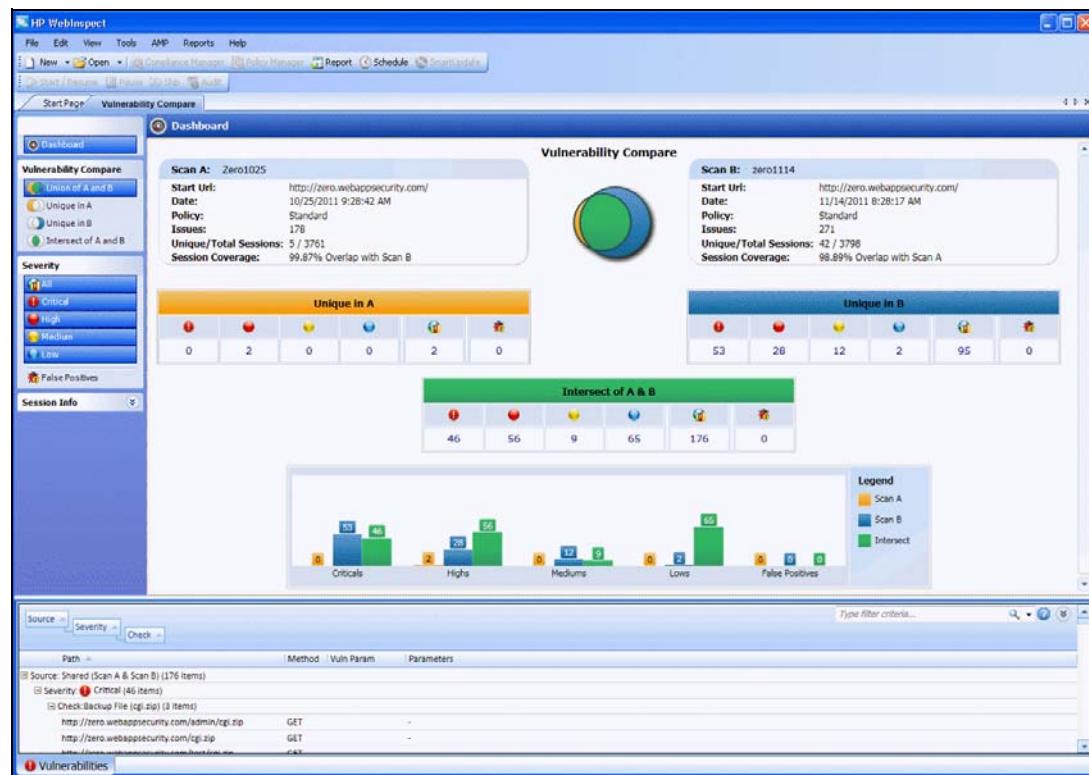
- Compare authorization access - Conduct scans using two different user accounts to discover vulnerabilities that are unique or common to both accounts.

➤ Note: Data from both scans must be stored in the same database type (SQL Server Express Edition vs. SQL Server Standard/Enterprise Edition).

Do one of the following to compare two scans:

- From the Manage Scans page, select two scans and click **Compare**.
- From a tab containing an open scan, click **Compare**; then select a scan from the list on the *Scan Comparison* dialog and click **Compare**. Note: If the open scan is a “site retest” (resulting from Rescan ® Verify Vulnerabilities), WebInspect automatically selects the parent scan for comparison. For example, if you created a scan named “zero,” and then verified vulnerabilities for that scan, the resulting scan would be named (by default) “site retest - zero.” With the retest scan open, if you select Compare, WebInspect will compare “site retest - zero” with the parent scan “zero.”

A warning message appears if the selected scans have different start URLs, if they used different scan policies, or if the scans are a different type (such as a Web site scan vs. a Web service scan). You can choose to continue, or you can terminate the function.



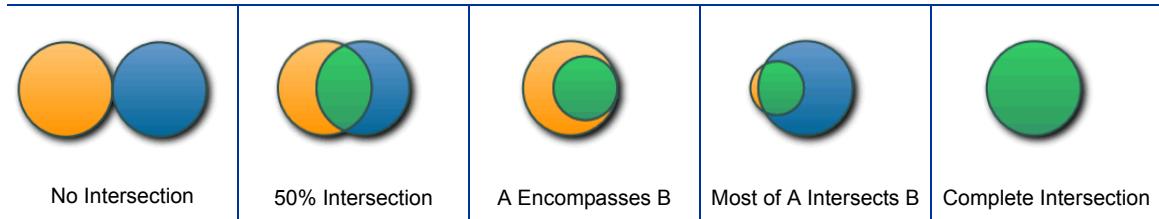
Dashboard

The dashboard of the **Vulnerability Compare** tab displays the following information for both Scan A and Scan B:

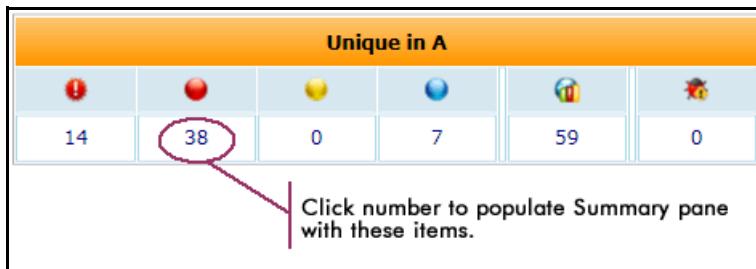
- Scan A/B: Name of the scan.
- Start Url: URL of the target site.
- Date: Date and time the original scan was conducted.

- Policy: Policy used for the scan; see [Appendix B, Policies and Components](#), for more information.
- Issues: Number of vulnerabilities and false positives detected.
- Unique/Total Sessions: Number of unique sessions created for this scan, compared to the total number of sessions.
- Session Coverage: Percentage of sessions that are common to both scans.

The dashboard also displays a Venn diagram depicting two sets: the vulnerabilities discovered by Scan A (represented by a yellow circle) and the vulnerabilities discovered by Scan B (represented by a blue circle). The intersection of the two sets is represented by the green overlap. The diagram is scaled to reflect the actual relationship between the sets.



The dashboard also displays the number of unique vulnerabilities, sorted by severity, detected by each individual scan and those detected by both scans (the intersection), as well as a bar chart representing those results.



Vulnerability Compare

To display in the Summary pane a list of vulnerabilities detected by one or both scans, select a filter from the Vulnerability Compare panel. The choices are:

- Union of A and B
- Unique in A
- Unique in B
- Intersection of A and B

You can also click the appropriate section of the Venn diagram to display vulnerabilities unique in A, unique in B, or those that are found in both A and B.

► Note: When comparing scans, WebInspect ignores the host and port. Consider two duplicate sites that are hosted on different servers. One site that is under development might be hosted at `http://dev.mysite.com` while that same site might be undergoing testing at `http://QA.mysite.com`. The host URL and port are NOT considered when comparing sessions and vulnerabilities between scans.

For example:

- Scan A: Session 1 is an http request to the url `http://qa.mysite.com/stuff/page/info.asp`. It has a vulnerability.
- Scan B: Session 1 is an http request to the url `http://dev.mysite.com:8080/stuff/page/info.asp`. It has the same vulnerability as Scan A, session 1.

When comparing scans, session 1 will appear as “Intersection of A and B.”

Severity

Select one or more options from the Severity panel to further filter the vulnerability list according to severity. Your selection is cumulative. For example, if you select **Critical** and **High**, then vulnerabilities in both categories are displayed in the Summary panel at the bottom of the window. If you select a third category (such as **Low**), vulnerabilities in all three categories are displayed. You can also select **All** (to display all vulnerabilities detected) or **False Positives** (to display only those sessions that WebInspect originally flagged as containing a vulnerability and which a user later determined were false positives).

To remove a category, click the highlighted category name.

Session Info

If you select (in the Summary pane) the URL of a session in which a vulnerability was detected, the Session Info panel becomes populated with choices you can use to investigate various attributes of the session. For all choices except **Vulnerability**, information for each scan is presented in separate panes. For specific information, see [Session Info Panel](#) on page 76.

Summary Pane

When comparing scans, use the horizontal summary pane at the bottom of the window to view a centralized display of vulnerable resources and quickly access vulnerability information. This pane lists the vulnerabilities belonging to the set you select in the Vulnerability Compare pane and the category of vulnerabilities you select in the Severity pane.

Note: You can also group and filter results. For more information, see [Using Filters and Groups in the Summary Pane](#) on page 88 .

Path	Method	Vuln Param	Parameters	Scan Name
http://zero.webappsecurity.com/forgot1.asp	GET	get	(Query)get...	Scan A & Scan B
http://zero.webappsecurity.com/forgot1.asp	GET	get	(Query)get...	Scan A & Scan B
http://zero.webappsecurity.com/join1.asp	GET		(Query)Na...	Scan A & Scan B
http://zero.webappsecurity.com/login1.asp	POST	login	(Post)login...	Scan A & Scan B
http://zero.webappsecurity.com/login1.asp	POST	login	(Post)login...	Scan A & Scan B
[+] Check:IIS ASP Chunked Encoding Overflow(4 items)				
http://zero.webappsecurity.com/forgot2.asp	POST		(Query)ms...	Scan A
http://zero.webappsecurity.com/pcomboindex.asp	POST		(Post)10PA...	Scan A
http://zero.webappsecurity.com/pformresults.asp	POST		(Post)10PA...	Scan A
http://zero.webappsecurity.com/plink.asp	POST		(Query)a=b...	Scan A
[+] Check:IIS Global Server Variables Disclosure (global.asa.bak)(1 item)				
http://zero.webappsecurity.com/global.asa.bak	GET		-	Scan A & Scan B
[+] Check:SQL Injection (' OR)(5 items)				
Vulnerabilities				

By default, the grid appears with the following columns: Path, Method, Vuln Param, and Scan Name. To add or delete columns, right-click the column header bar and choose **Columns** from the shortcut menu. The available columns are:

- Severity: A relative assessment of the vulnerability, ranging from low to critical. See below for associated icons.
- Check: A WebInspect probe for a specific vulnerability, such as cross-site scripting, unencrypted log-in form, etc.
- Path: The hierarchical path to the resource.
- Method: HTTP method, such as GET, PUT, etc.
- Stack: Stack trace information obtained from SecurityScope. This column is available only if SecurityScope was installed on the target server at the time of the scan.
- Vuln Param: The name of the vulnerable parameter.
- Parameters: Names of parameters and values assigned to them.
- Manual: Displays a check mark if the vulnerability was manually created.
- Duplicates: Vulnerabilities detected by SecurityScope that are traceable to the same source.
- Location: Path plus parameters.
- CWE ID: The Common Weakness Enumeration identifier(s) associated with the vulnerability.
- Kingdom: The category in which this vulnerability is classified, using a taxonomy of software security errors developed by the HP Fortify Software Security Research Group. Click the hyperlink to navigate to <http://www.hpenterprisesecurity.com/vulncat/en/vulncat/index.html>
- Scan Name: Scan A, Scan B, or Scan A & Scan B.

The severity of vulnerabilities is indicated by the icons illustrated in the following table.

Critical	High	Medium	Low

With a session selected, you can also view associated information by selecting an option from the Session Info panel.

For Post and Query parameters, click an entry in the Parameters column to display a more readable synopsis of the parameters.

Right-clicking an item in the Summary pane displays a shortcut menu containing the following commands:

- **Copy URL:** Copies the URL to the Windows clipboard.
- **Copy Selected Item(s):** Copies the text of selected items to the Windows clipboard.
- **Copy All Items:** Copies the text of all items to the Windows clipboard.
- **Export:** Creates a comma-separated values (csv) file containing either all items or selected items and displays it in Microsoft Excel.
- **View in Browser:** Renders the HTTP response in a browser.
- **Filter by Current Value:** Restricts the display of vulnerabilities to those that satisfy the criteria you select. For example, if you right-click on “Post” in the Method column and then select **Filter by Current Value**, the list displays only those vulnerabilities that were discovered by sending an HTTP request that used the Post method.

The filter criterion is displayed in the combo box in the upper right corner of the Summary pane. Alternatively, you can manually enter or select a filtering criterion using this combo box. For additional details and syntax rules, see [Using Filters and Groups in the Summary Pane](#) on page 88.

- **Review Vulnerability:** Allows you to retest the vulnerability. If the vulnerability was detected in only one scan, the *Vulnerability Review* window opens; if the vulnerability was detected in both scans, you are first prompted to select a scan. See [see Retesting/Reviewing Vulnerabilities](#) on page 85 for more information.

Note: The **Mark As** and **Send To** buttons are not enabled on the *Vulnerability Review* window.

- **Tools:** Presents a submenu of available tools.

FilesToURLs Utility

As part of the normal installation procedure, WebInspect installs two command line utilities (FilesToURLs.exe and FilesToURLs.py) designed to enhance the discovery and assessment of all resources on your Web site. When executed on your server, the utility examines all files on the target site and creates an XML file containing a URL for each resource it detects. Then, when using the new Web Site Scan Wizard, you can select the List-Driven scan method and submit this XML file to WebInspect.



FilesToURLs.exe requires .NET Framework 2.0 or later. FilesToURLs.py requires Python 2.6.

To create the XML file and include it in a scan:

- 1 Locate FilesToURLs.exe (or FilesToURLs.py, for UNIX systems). The default location is C:\Program Files\HP\HP WebInspect\.
- 2 Using a network share (or after copying the file to your Web server) run the utility, according to the usage described below.

- 3 Launch WebInspect.
- 4 On Step 1 of the Scan Wizard, select **List-Driven Scan**.
- 5 Click the Browse button and select the XML file generated by the FilesToURLs utility.
- 6 Complete the wizard and start the scan.

Usage for FilesToURLs.exe

FilesToURLs.exe /docroot c:\docroot /outfile outfile.xml [/include filename.xml] [/hostname example.com] [/baseurl baseurl] [/port port] [/secure] [/?] [/help]

Argument	Description
/docroot	The local path where web files are stored (required).
/outfile	The name of the XML file to be created (required).
/include	An existing file whose contents should be included in the output.
/hostname	The hostname from which files are served (default: local hostname).
/baseurl	The base URL from which files are served (default: /).
/port	The port that the web server is listening in (default: 80 or 443).
/secure	Specifies that the port is using SSL.

Usage for FilesToURLs.py

FilesToURLs.py [options]

Option	Description
-h, --help	Show help message and exit.
-d DOCROOT, --docroot=DOCROOT	Apache's DocumentRoot or other directory from which files are served.
-o FILE, --outfile=FILE	Write output to FILE (defaults to STDOUT).
-i FILE, --include=FILE	Include the contents of FILE in the output.
-n HOSTNAME, --hostname=HOSTNAME	Hostname of the web server (defaults to local hostname).
-b BASEURL, --baseurl=BASEURL	Base URL from which files are served (defaults to /).
-p PORT, --port=PORT	Port that service is listening on (defaults to 80 or 443).
-s, --secure	Specifies that the listening port is using SSL (defaults to False).

The List-Driven Scan option can also use a manually created plain text file instead of the XML file generated by the FilesToURLs utility. List one URL per line. Each URL must be fully qualified and must include the protocol (for example, http:// or https://).

Internet Protocol Version 6

WebInspect (beginning with version 8.1) supports Internet Protocol version 6 (IPv6) addresses in web site and web service scans. When you specify the Start URL, you must enclose the IPv6 address in brackets. For example:

- `http://[::1]`
WebInspect scans “localhost.”
- `http://[fe80::20c:29ff:fe32:bae1]/subfolder/`
WebInspect scans the host at the specified address starting in the “subfolder” directory.
- `http://[fe80::20c:29ff:fe32:bae1]:8080/subfolder/`
WebInspect scans a server running on port 8080 starting in “subfolder.”

5 Default Scan Settings

Introduction

To access this feature, click **Edit → Default Scan Settings**.

The default settings are divided into the following categories:

- Scan Settings
- Crawl Settings
- Audit Settings

When you click a category in the left pane, related information displays in the right pane.

Scan Settings

The parameters that control the manner in which WebInspect conducts a scan are available in the Scan Settings category.

Method

The Method settings broadly determine the type of scan to be conducted.

Scan Mode

Crawl Only

This option completely maps a site's tree structure. After a crawl has been completed, you can click **Audit** to assess an application's vulnerabilities.

Crawl and Audit

As WebInspect maps the site's hierarchical data structure, it audits each resource (page) as it is discovered (rather than crawling the entire site and then conducting an audit). This option is most useful for extremely large sites where the content may possibly change before the crawl can be completed. This is described in the Default Settings as Crawl and Audit (Simultaneously).

Audit Only

WebInspect applies the methodologies of the selected policy to determine vulnerability risks, but does not crawl the Web site. No links on the site are followed or assessed.

Manual

Manual mode allows you to navigate manually to whatever sections of your application you choose to visit, using Internet Explorer. It does not crawl the entire site, but records information only about those resources that you encounter while manually navigating the site. This feature is used most often to enter a site through a Web form logon page or to define a discrete subset or portion of the application that you want to investigate. Once you finish navigating through the site, you can audit the results to assess the security vulnerabilities related to that portion of the site that you recorded.

Crawl and Audit Mode

Simultaneously

As WebInspect maps the site's hierarchical data structure, it audits each resource (page) as it is discovered (rather than crawling the entire site and then conducting an audit). This option is most useful for extremely large sites where the content may possibly change before the crawl can be completed.

Sequentially

In this mode, WebInspect crawls the entire site, mapping the site's hierarchical data structure, and then conducts a sequential audit, beginning at the site's root.

If you select **Sequentially**, you can specify the order in which the crawl and audit should be conducted:

- **Test each engine type per session**—WebInspect audits all sessions using the first audit engine, then audits all sessions using the second audit engine, continuing in sequence until all engine types have been deployed.
- **Test each session per engine type**—WebInspect runs all audit engines against the first session, then runs all audit engines against the second session, continuing in sequence until all sessions are audited.

Navigation

Auto-fill Web forms during crawl

If you select this option, WebInspect submits values for input controls found on all forms. The values are extracted from a file you create using the Web form editor. Use the browse button to specify the file containing the values you want to use. Alternatively, you can click **Edit** (to modify the currently selected file) or **Create** (to create a Web form file).



Do not rely on this feature for authentication. If the crawler and the auditor are configured to share state, and if the scanner never inadvertently logs out of the site, then using values extracted by the Web Form Editor for a login form may work. However, if the audit or the crawl triggers a logout after the initial login, then the scanner will not be able to log in again and the auditing will be unauthenticated. To prevent WebInspect from terminating prematurely if it inadvertently logs out of your application, go to Scan Settings - Authentication and select **Use a login macro for forms authentication**.

Prompt for Web form values

If you select this option, WebInspect pauses the scan when it encounters an HTTP or JavaScript form and displays a window that allows you to enter values for input controls within the form. However, if you also select **Only prompt for tagged inputs**, WebInspect will not pause for user input unless a specific input control has been designated **Mark as Interactive Input** (using the Web Form Editor). This pausing for input is termed “interactive mode” and you can cancel it at any time during the scan.



Do not select this option if you want to conduct an unattended scan.

Use Web Service Design

This option applies only to Web Service scans.

When performing a Web services scan, WebInspect crawls the WSDL site and submits a value for each parameter in each operation. These values are contained in a file that you create using the Web Service Test Designer tool. WebInspect then audits the site by attacking each parameter in an attempt to detect vulnerabilities such as SQL injection.

Use the browse button to specify the file containing the values you want to use. Alternatively, you can click **Edit** (to modify the currently selected file) or **Create** (to create a SOAP values file).

General

Scan Details

Enable Path Truncation

Path truncation attacks are requests for known directories without file names. This may cause directory listings to be displayed. 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`, then truncating the path to look for `http://www.site.com/folder1/folder2/` and `http://www.site.com/folder1/` may cause the server to reveal directory contents or may cause unhandled exceptions.

Attach debug information in request header

If you select this option, WebInspect includes a “Memo:” header in the HTTP request containing information that can be used by support personnel to diagnose problems. Although the format and content is subject to change without notice, the information may assist advanced users. Two of the more useful constructions are illustrated below.

Attack memo header example:

```
Memo: 197:Auditor.SendAsynchronousRequest:Attack(CID:123:AS:2,  
EID:1354e211-9d7d-4cc1-80e6-4de3fd128002,ST:AuditAttack,AT:PostParamManip  
ulation,APD:username,I:(1,0),R:False,SM:2,SID:FDF074B3AC41D4ABE4114B3C1A1  
14160,PSID:DDAA45FB26C9149DB15AF2D8DDFD5D3A)  
Requestor thread ID handling request:197  
Originating function in scanner: SendAsynchronousRequest  
CheckID:123  
Attack Sequence: 2  
Originating Engine ID:1354e211-9d7d-4cc1-80e6-4de3fd128002
```

```
Session Type: AuditAttack
Attack Type: PostParamManipulation
Attack descriptor (what was attacked): username 'param' was attacked; it
is parameter (1,0) in collection
Smart Mode: 2
Attack Session ID: FDF074B3AC41D4ABE4114B3C1A114160
Parent Session ID :DDAA45FB26C9149DB15AF2D8DDFD5D3A
```

Crawl memo header example:

```
Memo: 180:ProcessSession:Crawler.CreateStateRequest:
SID:2BC3FC705779A6F201810A1E64F7CF83,PSID:A77674B6A5BF9B3B3CEDAEF583C0826
2,ST:Crawl,CLT:HTML
Requestor thread ID handling request:180
Originating function in scanner: ProcessSession:Crawler.CreateStateRequest
Session Type: Crawl
Crawl Link Type: HTML
Session ID: 2BC3FC705779A6F201810A1E64F7CF83
Parent Session ID : A77674B6A5BF9B3B3CEDAEF583C0826
```

Case-sensitive request and response handling

Select this option if the server at the target site is case-sensitive to URLs. Usually, the case sensitivity of a Web server is determined by the server's operating system. Windows is not case-sensitive; UNIX and Linux are. The one exception to this rule concerns Apache, which can be configured with non-case-sensitive page names, even on a UNIX system.

Recalculate correlation data

This feature is used only for comparing scans and should be selected only upon the advice of HP Support personnel if scan comparisons produce questionable results.

Compress response data

If you select this option, WebInspect saves disk space by storing each HTTP response in a compressed format in the database.

Enable Traffic Monitor Logging

While scanning, WebInspect displays in the navigation pane only those sessions that reveal the hierarchical structure of the Web site plus those sessions in which a vulnerability was discovered. However, if you select the Traffic Monitor option, WebInspect adds the **Traffic Monitor** button to the Scan Info panel (as shown below), allowing you to display and review every request sent by WebInspect and the associated response received from the server.

Time	Host	Method	Uri	Response Code	Engine
1/6/2011 2:53:36 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:37 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:53:53 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:54:14 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:54:14 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...
1/6/2011 2:54:15 PM	h10078.www1.hp.com	POST	/cda/hpdc/register.do	200 OK	Cross Site...

Encrypt Traffic Monitor File

All sessions are normally recorded in the traffic monitor file as clear text. If you are concerned about storing sensitive information such as passwords on your computer, you can elect to encrypt the file.

- Encrypted files cannot be compressed. Selecting this option will significantly increase the size of exported scans containing log files.

Maximum crawl-audit recursion depth

When an attack reveals a vulnerability, WebInspect crawls that session and follows any link that may be revealed. If that crawl and audit reveals a link to yet another resource, the depth level is incremented and the discovered resource is crawled and audited. This process can be repeated until no other links are found. However, to avoid the possibility of entering an endless loop, you may limit the number of recursions. The default value is 2; the maximum recursion level is 1,000.

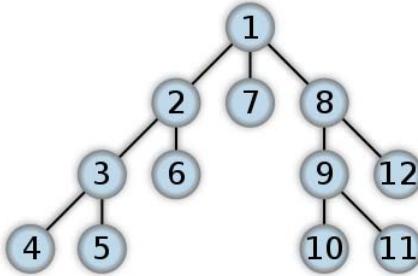
Crawl Details

Crawler

Select either **Depth First** or **Breadth First**.

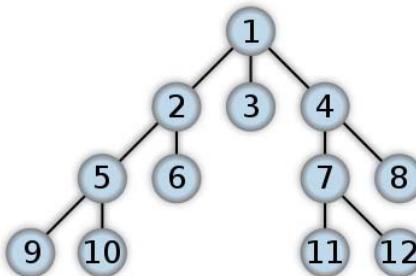
Depth-first crawling accommodates sites that enforce order-dependent navigation (where the browser must visit page A before it can visit page B). This type of search progresses by expanding the first child node (link) and crawling deeper and deeper until it reaches a node that has no children. The search then backtracks, returning to the most recent node it hasn't

finished exploring and drilling down from there. The following illustration depicts the order in which linked pages are accessed using a depth-first crawl. Node 1 has links to nodes 2, 7, and 8. Node 2 has links to nodes 3 and 6.



Depth-First Tree

By contrast, breadth-first crawling begins at the root node and explores all the neighboring nodes (one level down). Then for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until all resources are identified. The following illustration depicts the order in which linked pages are accessed using a breadth-first crawl. Node 1 has links to nodes 2, 3, and 4. Node 2 has links to nodes 5 and 6.



Breadth-First Tree

When performing a depth-first crawl, WebInspect pursues links in a fashion that more closely represents human interaction. While slower than breadth-first crawling, the depth-first method accommodates applications that enforce ordering of requests (such as requiring the user to visit a “shopping cart” page before accessing the “check-out” page).

Enable keyword search audit

A keyword search, as its name implies, examines server responses and looks for certain text strings that typically indicate a vulnerability. This option is available only for a crawl-only scan.

Perform redundant page detection

Highly dynamic sites could create an infinite number of resources (pages) that are virtually identical. If allowed to pursue each resource, WebInspect would never be able to finish the scan. This option, however, allows WebInspect to identify and exclude processing of redundant resources.

Limit maximum single URL hits to

Sometimes, the configuration of a site will cause a crawl to loop endlessly through the same URL. Use this field to limit the number of times a single link will be followed during a crawl. The default value is 5.

Include parameters in hit count

If you select **Limit maximum single URL hits to** (above), a counter is incremented each time the same URL is encountered. However, if you also select **Include parameters in hit count**, then when parameters are appended to the URL specified in the HTTP request, the crawler will crawl that resource up to the single URL limit. Any differing set of parameters is treated as unique and has a separate count.

For example, if this option is selected, then “page.aspx?a=1” and “page.aspx?b=1” will both be counted as unique resources (meaning that the crawler has found two pages).

If this option is not selected, then “page1.aspx?a=1” and “page.aspx?b=1” will be treated as the same resource (meaning that the crawler has found the same page twice).

Limit maximum link traversal sequence

This option restricts the number of hyperlinks that can be sequentially accessed as WebInspect crawls the site. For example, if five resources are linked as follows

Page A contains a hyperlink to Page B

Page B contains a hyperlink to Page C

Page C contains a hyperlink to Page D

Page D contains a hyperlink to Page E

and if this option is set to “3,” then Page E will not be crawled. The default value is 15.

Limit maximum crawl folder depth to

This option limits the number of directories that may be included in a single request. For example, if the URL is

`http://www.mysite.com/Dir1/Dir2/Dir3/Dir4/Dir5/Dir6/Dir7`

and this option is set to “4,” then the contents of directories 5, 6, and 7 will not be crawled. The default value is 15.

Limit maximum crawl count to

This feature restricts the number of HTTP requests sent by the crawler and should be used only if you experience problems completing the scan of a large site.

Limit maximum Web form submissions to

Normally, when WebInspect encounters a form that contains controls having multiple options (such as a list box), it extracts the first option value from the list and submits the form; it then extracts the second option value and resubmits the form, repeating this process until all option values in the list have been submitted. This ensures that all possible links will be followed.

There are occasions, however, when submitting the complete list of values would be counterproductive. For example, if a list box named “State” contains one value for each of the 50 states in the United States, there is probably no need to submit 50 instances of the form.

Use this setting to limit the total number of submissions that WebInspect will perform. The default value is 3.

Audit Details

If you select a depth-first crawl, you can also elect to retrace the crawl path for each parameter attack, as opposed to applying all attacks as the crawl progresses. This considerably increases the time required to conduct a scan. However, this option should be used for sites that enforce a strict order of access to pages. Using the “depth first” and “path retrace” options can obtain a successful scan on these types of sites when a breadth-first crawl fails.

Content Analyzers

Flash

If you enable the Flash analyzer, WebInspect analyzes Flash files, Adobe’s vector graphics-based resizable animation format.

JavaScript/VBScript

The JavaScript/VBScript analyzer is always enabled. It allows WebInspect to crawl links defined by JavaScript or VisualBasic script, and to create and audit any documents rendered by JavaScript.

To increase the speed at which WebInspect conducts a crawl while analyzing script, change your browser options so that images/pictures are not displayed.

Configure the settings described below.

Crawl links found from script execution

If you select this option, the crawler will follow dynamic links (i.e., links generated during JavaScript execution).

Reject script include file requests to offsite hosts

Pages downloaded from a server may contain scripts that retrieve files and dynamically render their content. An example JavaScript “include file” request is:

```
<script type="text/javascript" src="www.badsite.com/yourfile.htm"></script>
```

WebInspect will download and parse such files, regardless of their origin or file type, unless you select the Reject Script option. It will then download the files only if permitted by the parameters normally governing file handling (such as session and attack exclusions, allowed hosts, etc.).

Isolate script analysis (out-of-process execution)

WebInspect analyzes and executes JavaScript and VBScript to discover links to other resources. Applications or Web sites containing an inordinate number of links can sometimes exhaust the amount of memory allocated to this process. If this occurs, you can assign this function to a separate (remote) process, which will accommodate an infinite number of links. You may, however, notice a slight increase in the amount of time required to scan the site.

Create script event sessions

If you select this option, WebInspect creates and saves a session for each change to the Document Object Model (DOM).

Verbose script parser debug logging

If you select this setting AND if the Application setting for logging level is set to Debug, WebInspect logs every method called on the DOM object. This can easily create several gigabytes of data for medium and large sites.

Log JavaScript errors

WebInspect logs JavaScript parsing errors from the script parsing engine.

Max script events per page

Certain scripts endlessly execute the same events. You can limit the number of events allowed on a single page to a value between 1 and 9999. The default value is 1000.

Enable JavaScript UI Exclusions

If selected, the WebInspect JavaScript parser ignores JQuery calendars.

Silverlight

If you enable the Silverlight analyzer, WebInspect analyzes Silverlight applications, which provide functionalities similar to those in Adobe Flash, integrating multimedia, graphics, animations and interactivity into a single runtime environment.

Recommendations

While conducting a scan, WebInspect may encounter certain omissions, abnormalities, or anomalies that interfere with or diminish the thoroughness of the scan. If you enable the Recommendations feature, WebInspect records this information and, when the scan is complete (or paused), presents a list of recommendations designed to improve the quality of your scan when you next conduct it.

To enable this feature, select **Run Recommendation Modules when the scan is paused or completed**. You can then select or deselect an individual module by selecting or clearing its associated check box.

To view the recommendations resulting from this analysis, click **Recommendations** in the **Scan Info** panel.

Network Authentication

This module detects that network, proxy, or site authentication is required, but credentials are missing or invalid.

Web Macro

This module tracks the number of times the scanner runs a Web macro to log in to the site and warns if the number seems to be excessive, indicating that the macro may not be functioning properly. Usually this occurs because the macro is unable to log in or contains a poor log-out condition, or the site prevents multiple concurrent log-in sessions.

Note: If the macro worked correctly when it was recorded, the user name or password assigned to the site may have been subsequently changed, or the account may have been blocked or deleted.

The threshold for determining this condition is heuristically set at 10 percent. For example, if the scanner examines 4,000 responses and more than 400 of them (10 percent) indicate that the scanner is logged out of the site, thus causing the scanner to run the macro that logs in to the site, then there is a high probability that the macro is faulty and should be replaced.

You may establish a threshold that is higher or lower than 10 percent, based on your experience. To do so, click **Settings** and select a different macro ratio.

File Not Found

This module examines the server's responses to requests for files and determines that the scan settings for recognizing a "file not found" condition may be incorrect. It is used only during a crawl-and-audit scan.

Web Service

This module detects the presence of Web service communication within the Web site and advises you to conduct a Web service scan.

Form Values

This module detects the existence of forms containing an input element for which you have not provided a value.

Caution: Using Form Values recommendations can cause an unintended large increase in scan data stored, as well as potential "out of memory" errors during large scans. This module is turned off by default. If it is turned on, data storage problems and out of memory incidents may occur.

Custom Parameters

This module detects the use of URL Rewriting techniques and RESTful services technologies.

Click **Settings** to select options for this feature. The settings are:

- **Similar URLs Percentage of Total in Site** -- In some cases, WebInspect uses a "Similar URLs Percentage of Total in Site" threshold to prevent false positives. You can change this threshold to improve the accuracy of these suggestions in case you encounter false positives. Enter a percentage between 1 and 100.
- **Maximum Custom Parameter Recommendations** -- WebInspect also applies a prioritization algorithm to the recommendations and lists them in order of their estimated accuracy. If you encounter problems or anomalies while attempting to detect custom parameters, you can use the Maximum Custom Parameter Recommendations box to limit the number of recommendations the module will produce. Select either Unlimited, or enter a value between 1 and 10,000.
- **Validate Custom Parameter Recommendations** -- If you select this option, WebInspect validates custom parameter recommendations by sending appropriate requests during the analysis stage and removing those recommendations for which an invalid response is received.

Requestor

A requestor is the software module that handles HTTP requests and responses.

Requestor Performance

Use a shared requestor

If you select this option, the crawler and the auditor use a common requestor when scanning a site, and each thread uses the same state, which is also shared by both modules. This replicates the technique used by previous versions of WebInspect and is suitable for use when maintaining state is not a significant consideration. You also specify the maximum number of threads.

Use separate requestors

If you select this option, the crawler and auditor use separate requestors. Also, the auditor's requestor associates a state with each thread, rather than having all threads use the same state. This method results in significantly faster scans.

You can also specify the maximum number of threads that can be created for each requestor. When performing a sequential crawl and audit, the crawl requestor can be configured to send up to 25 concurrent HTTP requests before waiting for an HTTP response to the first request; the maximum for the audit requestor is 50. The default setting is 5 for the crawl requestor and 10 for the audit requestor. Increasing the thread count will increase the speed of a scan, but might also exhaust your system resources as well as those of the server you are scanning.

- If you select “simultaneous crawl and audit” as the [Scan Mode](#), the Crawl Requestor Thread Count is set to “1” and may not be modified.

If you notice numerous entries on the [Scan Log](#) tab showing requests timing out, you should reduce the thread count. While most servers can handle a large number of requests, servers in development environments sometimes have limitations on their licensing that allow only five or fewer users to be connected at a single time. In such cases, you should reduce the thread count to be less than 5. Failing to do so may mean that WebInspect does not accurately crawl or audit the site because requests are being rejected by the server.

Requestor Settings

Limit maximum response size to

Select this option to limit the size of accepted server responses, and then specify the maximum size (in kilobytes). The default is 1000 kilobytes. Note that Flash files (.swf) and JavaScript “include” files are not subject to this limitation.

Request retry count

Specify how many times WebInspect will resubmit an HTTP request after receiving a “failed” response (which is defined as any socket error or request timeout). The value must be greater than zero.

Request timeout

Specify how long WebInspect will wait for an HTTP response from the server. If this threshold is exceeded, WebInspect resubmits the request until reaching the retry count. If it then receives no response, WebInspect logs the timeout and issues the first HTTP request in the next attack series. The default value is 20 seconds.

Stop Scan if Loss of Connectivity Detected

There may be occasions during a scan when a Web server crashes or becomes too busy to respond in a timely manner. You can instruct WebInspect to terminate a scan by specifying a threshold for the number of timeouts.

Consecutive “single host” retry failures to stop scan

Enter the number of consecutive timeouts permitted from one specific server. The default value is 75.

Consecutive “any host” retry failures to stop scan

Enter the total number of consecutive timeouts permitted from all hosts. The default value is 150.

Nonconsecutive “single host” retry failures to stop scan

Enter the total number of nonconsecutive timeouts permitted from a single host. The default value is “unlimited.”

Nonconsecutive “any host” retry failures to stop scan

Enter the total number of nonconsecutive timeouts permitted from all hosts. The default value is 350.

If first request fails, stop scan

Selecting this option will force WebInspect to terminate the scan if the target server does not respond to WebInspect’s first request.

Response codes to stop scan if received

Enter the HTTP status codes that, if received, will force WebInspect to terminate the scan. Use a comma to separate entries; use a hyphen to specify an inclusive range of codes.

Session Storage

Log Rejected Session to Database

You can specify which rejected sessions should be saved to the WebInspect database. This saved information can be used for two purposes.

- If you pause a scan, change any of the settings associated with the Reject Reasons in this panel, and then resume the scan, WebInspect retrieves the saved data and sends HTTP requests that previously were suppressed.
- Hewlett-Packard support personnel can extract the generated (but not sent) HTTP requests for analysis.

Sessions may be rejected for the reasons cited in the following table:

Reasons for Rejecting a Session

Reject Reason	Explanation
Invalid Host	Any host that is not specified in Default (or Current) Scan Settings/Scan Settings/Allowed Hosts.
Excluded File Extension	Files having an extension that is excluded by settings specified in Default (or Current) Scan Settings/Scan Settings/Session Exclusions/Excluded or Rejected File Extensions; also Default (or Current) Scan Settings/Crawl Settings/Session Exclusions/Excluded or Rejected File Extensions; also Default (or Current) Scan Settings/Audit Settings/Session Exclusions/Excluded or Rejected File Extensions.
Excluded URL	URLs or hosts that are excluded by settings specified in Default (or Current) Scan Settings/Scan Settings/Session Exclusions/Excluded or Rejected URLs and Hosts; also Default (or Current) Scan Settings/Crawl Settings/Session Exclusions/Excluded or Rejected URLs and Hosts; also Default (or Current) Scan Settings/Audit Settings/Session Exclusions/Excluded or Rejected URLs and Hosts.
Outside Root URL	If the Restrict to Folder option is selected when starting an advanced scan, any resource not qualified by the available options (Directory Only, Directory and Subdirectories, or Directory and Parent Directories).
Maximum Folder Depth Exceeded	HTTP requests were not sent because the value specified by the Limit maximum crawl folder depth to option in Default (or Current) Scan Settings/Scan Settings/General has been exceeded.
Maximum URL Hits	HTTP requests were not sent because the value specified by the Limit Maximum Single URL hits to option in Default (or Current) Scan Settings/Scan Settings/General has been exceeded.
404 Response Code	In the Default (or Current) Scan Settings/Scan Settings/File Not Found group, the option Determine File Not Found (FNF) using HTTP response codes is selected and the response contains a code that matches the requirements.

Reasons for Rejecting a Session (cont'd)

Reject Reason	Explanation
Solicited File Not Found	In the Default (or Current) Scan Settings/Scan Settings/File Not Found group, the option Auto detect FNF page is selected and WebInspect determined that the response constituted a "file not found" condition.
Custom File Not Found	In the Default (or Current) Scan Settings/Scan Settings/File Not Found group, the option Determine FNF from custom supplied signature is selected and the response contains one of the specified phrases.
Rejected Response	Files having a MIME type that is excluded by settings specified in Default (or Current) Scan Settings/Scan Settings/Session Exclusions/Excluded MIME Types; also Default (or Current) Scan Settings/Crawl Settings/Session Exclusions/Excluded MIME Types; also Default (or Current) Scan Settings/Audit Settings/Session Exclusions/Excluded MIME Types.

Session Storage

WebInspect normally saves only those attack sessions in which a vulnerability was discovered. To save all attack sessions, select **Save non-vulnerable attack sessions**.

Session Exclusions

These settings apply to both the crawl and audit phases of a WebInspect vulnerability scan. To specify exclusions for only the crawl or only the audit, use the **Crawl Settings - Session Exclusions** or the **Audit Settings - Sessions Exclusions** settings.

Excluded or Rejected File Extensions

You can identify a file type and then specify whether you want to exclude or reject it.

- **Reject**—WebInspect will not request files of the type you specify.
- **Exclude**—WebInspect will request the files, but will not attack them (during an audit) and will not examine them for links to other resources.

By default, most image, drawing, media, audio, video, and compressed file types are rejected.

Follow the steps below to add a file extension:

- 1 Click **Add**.
The *Exclusion Extension* window opens.
- 2 In the **File Extension** box, enter a file extension.
- 3 Select either **Reject**, **Exclude**, or both.
- 4 Click **OK**.

Excluded MIME Types

WebInspect will not process files associated with the MIME type you specify.

Follow the steps below to add a MIME Type:

1 Click **Add**.

The *Provide a Mime-type to Exclude* window opens.

2 In the **Exclude Mime-type** box, enter a MIME type.

3 If you enter a regular expression to specify a MIME type, select the **Use Regular Expression** check box.

4 Click **OK**.

Other Exclusion/Rejection Criteria

You can identify various components of an HTTP message and then specify whether you want to exclude or reject a session that contains that component.

- **Reject**—WebInspect will not send any HTTP requests to the host or URL you specify. For example, you should usually reject any URL that deals with logging off the site, since you don't want to log out of the application before the scan is completed.
- **Exclude**—During a crawl, WebInspect will not examine the specified URL or host for links to other resources. During the audit portion of the scan, WebInspect will not attack the specified host or URL. If you want to access the URL or host without processing the HTTP response, select the **Exclude** option, but do not select **Reject**. For example, to check for broken links on URLs that you don't want to process, select only the **Exclude** option.

To edit the default criteria:

1 Select a criterion and click **Edit** (on the right side of the **Other Exclusion/Rejection Criteria** list).

The *Reject or Exclude a Host or URL* window opens.

2 Select either **Host** or **URL**.

3 In the **Host/URL** box, enter a URL or fully qualified host name, or a regular expression designed to match the targeted URL or host.

4 Select either **Reject**, **Exclude**, or both.

5 Click **OK**.

To add exclusion/rejection criteria:

1 Click **Add** (on the right side of the **Other Exclusion/Rejection Criteria** list).

The *Create Exclusion* window opens.

2 Select an item from the **Target** list.

3 If you selected **Query Parameter** or **Post Parameter** as the target, enter the **Target Name**.

4 From the **Match Type** list, select the method to be used for matching text in the target:

- **Matches Regex**: Matches the regular expression you specify in the **Match String** box.
- **Matches Regex Extension**: Matches a syntax available from HP's regular expression extensions you specify in the **Match String** box.
- **Matches**: Matches the text string you specify in the **Match String** box.
- **Contains**: Contains the text string you specify in the **Match String** box.

- 5 In the **Match String** box, enter the string or regular expression for which the target will be searched. Alternatively, if you selected a regular expression option in the **Match Type**, you can click the drop-down arrow and select **Create Regex** to launch the Regular Expression Editor.

6 Click .

7 (Optional) Repeat steps 2-6 to add more conditions. Multiple matches are ANDed.

8 If you are working in Current Settings, you can click **Test** to process the exclusions on the current scan. Any sessions from that scan that would have been filtered by the criteria will appear in the test screen, allowing you to modify your settings if required.

9 Click **OK**.

10 When the exclusion appears in the **Excluded or Rejected URLs or Hosts** list, select either **Reject**, **Exclude**, or both.

Example 1

To ensure that you ignore and never send requests to any resource at Microsoft.com, enter the following exclusion and select **Reject**.

Target	Target Name	Match Type	Match String
URL	n/a	contains	Microsoft.com

Example 2

Enter “logout” as the match string. If that string is found in any portion of the URL, the URL will be excluded or rejected (depending on which option you select). Using the “logout” example, WebInspect would exclude or reject URLs such as logout.asp or applogout.jsp.

Target	Target Name	Match Type	Match String
URL	n/a	contains	logout

Example 3

The following example rejects or excludes a session containing a query where the query parameter “username” equals “John.”.

Target	Target Name	Match Type	Match String
Query parameter	username	matches	John

Example 4

The following example excludes or rejects the following directories:

<http://www.test.com/W3SVC55/>

<http://www.test.com/W3SVC5/>

<http://www.test.com/W3SVC550/>.

Target	Target Name	Match Type	Match String
URL	n/a	matches regex	/W3SVC[0-9]*/

Allowed Hosts

Use the Allowed Host settings to add domains to be crawled and audited. If your Web presence uses multiple domains, add those domains here. For example, if you were scanning “WIexample.com,” you would need to add “WIexample2.com” and “WIexample3.com” here if those domains were part of your Web presence and you wanted to include them in the crawl and audit.

You can also use this feature to scan any domain whose name contains the text you specify. For example, suppose you specify www.myco.com as the scan target and you enter “myco” as an allowed host. As WebInspect scans the target site, if it encounters a link to any URL containing “myco,” it will pursue that link and scan that site’s server, repeating the process until all linked sites are scanned. For this hypothetical example, WebInspect would scan the following domains:

- www.myco.com:80
- contact.myco.com:80
- www1.myco.com
- ethics.myco.com:80
- contact.myco.com:443
- wow.myco.com:80
- mycocorp.com:80
- www.interconnection.myco.com:80

Note that if you specify a port number, then the allowed host must be an exact match.

To add allowed domains:

- 1 Click **Add**.
- 2 On the *Specify Allowed Host* window, enter a URL (or a regular expression representing a URL) and click **OK**.

When specifying the URL, do not include the protocol designator (such as http:// or https://).

To edit or remove an allowed domain:

- 1 Select a domain from the **Allowed Hosts** list.
- 2 Click **Edit** or **Remove**.

HTTP Parsing

HTTP Parameters Used for State

If your application uses URL rewriting or post data techniques to maintain state within a Web site, you must identify which parameters are used. For example, a PHP4 script can create a constant of the session ID named SID, which is available inside a session. By appending this to the end of a URL, the session ID becomes available to the next page. The actual URL might look something like the following:

```
.../page7.php?PHPSESSID=4725a759778d1be9bdb668a236f01e01
```

Because session IDs change with each connection, an HTTP request containing this URL would create an error when you tried to replay it. However, if you identify the parameter (PHPSESSID in this example), then WebInspect will replace its assigned value with the new session ID obtained from the server each time the connection is made.

Similarly, some state management techniques use post data to pass information. For example, the HTTP message content may include userid=slbhkelvbk173dhj. In this case, “userid” is the parameter you would identify.

 You need to identify parameters only when the application uses URL rewriting or posted data to manage state. It is not necessary when using cookies.

WebInspect can identify potential parameters if they occur as posted data or if they exist within the query string of a URL. However, if your application embeds session data in the URL as extended path information, you must provide a regular expression to identify it. In the following example, “1234567” is the session information:

```
http://www.onlinestore.com/bikes/(1234567)/index.html
```

The regular expression for identifying the parameter would be: /\([\w\d]+\)/

Determine State from URL Path

If your application determines state from certain components in the URL path, select this check box and add one or more regular expressions that identify those components. The defaults identify ASP.NET cookieless session IDs.

HTTP Parameters Used for Navigation

Some sites contain only one directly accessible resource, and then rely on query strings to deliver the requested information, as in the following examples:

Ex. 1 -- http://www.anysite.com?Master.asp?Page=1

Ex. 2 -- http://www.anysite.com?Master.asp?Page=2;

Ex. 3 -- http://www.anysite.com?Master.asp?Page=13;Subpage=4

Ordinarily, WebInspect would assume that these three requests refer to identical resources and would conduct a vulnerability scan on only one of them. Therefore, if your target Web site employs this type of architecture, you must identify the specific resource parameters that are used.

Examples 1 and 2 contain one resource parameter: “Page.”

Example 3 contains two parameters: “Page” and “Subpage.”

To identify resource parameters:

- 1 Click **Add**.

- 2 On the *HTTP Parameter* window, enter the parameter name and click **OK**.
The string you entered appears in the **Parameter** list.
- 3 Repeat this procedure for additional parameters.

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 WebInspect should use.

Custom Parameters

Custom Parameters are used to accommodate sites that use URL rewriting techniques and/or Representation State Transfer (REST) web services technologies. You can write rules for these custom parameters, or you can import rules from a common configuration file written in Web Application Description Language (WADL). In addition to applying these rules that you discretely define or import, WebInspect will attempt (during a scan) to identify custom parameters and create rules to accommodate them. WebInspect will save these rules in the Custom Parameters settings and will suggest them as recommendations.

URL Rewriting

Many dynamic sites use URL rewriting because static URLs are easier for users to remember and are easier for search engines to index the site. For example, an HTTP request such as

`http://www.pets.com>ShowProduct/7`

is sent to the server's rewrite module, which converts the URL to the following:

`http://www.pets.com>ShowProduct.php?product_id=7`

In this example, the URL causes the server to execute the php script "ShowProduct" and display the information for product number 7.

When WebInspect scans a page, it must be able to determine which elements are variables so that its attack agents can thoroughly check for vulnerabilities. To enable this, you must define rules that identify these elements. You can do so using either Simplified Syntax or Regular Expression syntax.

Examples:

HTML: `User 1 details`
Rule: `http://samplesite.com/someDetails/{username}/`

HTML: `User 1 details`
Rule: `http://samplesite.com/TwoParameters/Details/{username}/{parameter2}`

HTML: `User 1 details`
Rule: `http://samplesite.com/{parameter2}/PreFixParameter/Details/{username}`

RESTful Services

A RESTful web service (also called a RESTful web API) is a simple Web service implemented using HTTP and the principles of REST. It has gained widespread acceptance across the Web as a simpler alternative to SOAP- and Web Services Description Language (WSDL)-based Web services.

The following request adds a name to a file using an HTTP query string.

```
GET /adduser?name=Robert HTTP/1.1
```

This same function could be achieved by using the following method with a Web service. Note that the parameter names and values have been moved from the request URI and now appear as XML tags in the request body.

```
POST /users HTTP/1.1
Host: myserver
Content-Type: application/xml
<?xml version="1.0"?>
<user> <name>Robert</name>
</user>
```

In the case of both URL rewriting and RESTful web services, you must create rules that instruct WebInspect how to create the appropriate requests.

To create a rule:

- 1 Click **New Rule**.
- 2 In the Expression column, enter a rule. See [Creating Rules for Matrix and Path Parameters](#) on page 185 for guidelines and examples.

The enabled check box is selected by default. WebInspect examines the rule and, if valid, removes the red **X**.

To delete a rule:

- 1 Select a rule from the **Custom Parameters Rules** list.
- 2 Click **Delete**.

To disable a rule without deleting it:

- 1 Select a rule.
- 2 Clear the check mark in the **Enabled** column.

To import a file containing rules:

- 1 Click Import 

- 2 Using a standard file-selection dialog, select the type of file (.wadl or .txt) containing the custom rules you want to apply.
- 3 Locate the file and click **Open**.

Enable automatic seeding of rules that were not used during scan

The most reliable rules for custom parameters are those deduced from a WADL file or created by developers of the Web site. If a rule is not invoked during a scan (because the rule doesn't match any URL), then WebInspect can programmatically assume that a valid portion of the site has not been attacked. Therefore, if you select this option, WebInspect will create sessions to exercise these unused rules in an effort to expand the attack surface.

Double Encode URL Parameters

Double-encoding is an attack technique that encodes user request parameters twice in hexadecimal format in an attempt to bypass security controls or cause unexpected behavior from the application. For example, a cross-site scripting (XSS) attack might normally appear as:

```
<script>alert('FOO')</script>
```

This malicious code could be inserted into a vulnerable application, resulting in an alert window with the message “FOO.” However, the web application can have a filter that prohibits characters such as < (less than) > (greater than) and / (forward slash), since they are used to perform Web application attacks. The attacker could attempt to circumvent this safeguard by using a “double encoding” technique to exploit the client’s session. The encoding process for this Javascript is:

Char	Hex encode	Encoded % Sign	Double encoded result
<	%3C	%25	%253C
/	%2F	%25	%252F
>	%3E	%25	%253E

Finally, the malicious code, double-encoded, is:

```
%253Cscript%253Ealert('XSS')%253C%252Fscript%253E
```

If you select this option, WebInspect will create double-encoded URL parameters (instead of single-encoded parameters) and submit them as part of the attack sequence. This is recommended when the Web server uses, for example, Apache mod-rewrite plus PHP or Java URL Rewrite Filter 3.2.0.

Creating Rules for Matrix and Path Parameters

There are three ways rules can be created in the system. Rules may be:

- Entered manually.
- Generated from a WADL file specified by the user or received through SecurityScope.
- Imported from a flat file containing a list of rules.

When entering rules manually, you specify the path segments of a URL that should be treated as parameters.

The rules are actually modified URLs that use special characters to designate parts of the actual URL that contain parameters. If URL matches a rule, WebInspect parses the parameters and attacks them. Notable components of a rule (and of a URL) are:

- Scheme (HTTP/HTTPS)
- Authority (username + hostname + port)
- Path (gp/c/{book_name}/)
- Query (anything that follows “?”)
- Fragment (anything that follows “#”)

Definition of Path Segment

A path segment starts with ‘/’ characters and is terminated either by another ‘/’ character or by end of line. To illustrate, path “/a” has one segment whereas path “/a/” has two segments (the first containing the string “a” and the second being empty. Note that paths “/a” and “/a/” are not equal. When attempting to determine if a URL matches a rule, empty segments are considered.

Special Elements for Rules

A rule may contain the following special elements.

- * (Asterisk) May appear in production defined below; presence in non-path productions means that this part of the URL will not participate in matching (or, in other words, will match anything).
- {...} (Group) A named parameter that may appear within the path of the rule. The content has no special meaning and is used during reporting as the name of the attacked parameter. The character set allowed within a group is defined in RFC 3986 as *pchar:
 - pchar = unreserved / pct-encoded / sub-delims / ":" / "@"
 - pct-encoded = "%" HEXDIG HEXDIG
 - unreserved = ALPHA DIGIT - . _ ~
 - reserved = gen-delims / sub-delims
 - gen-delims = : / ? # [] @ "
 - sub-delims = ! \$ & ' () * + , ; =

A group’s content cannot include the “open bracket” and “close bracket” characters, unless escaped as pct-encoded element.

The rules for placing * out of path are described below. Within a path segment, any amount of * and {} groups can be placed, provided they’re interleaved with plain text. For example:

Valid rule: `http://www.amazon.com/gp/c/*={param}`

Invalid rule: `http://www.amazon.com/gp/c/*{}`

Rules with segments having **, *{}, {*} or {}{} entries are invalid.

For a rule to match a URL, all components of the rule should match corresponding components of the crawled URL. Path comparison is done segment-wise, with * and {} groups matching any number of characters (including zero characters), plain text elements matching corresponding plain text elements of the path segment of the URL. So, for example:

`http://www.amazon.com/gp/c/{book_name}` is a match for these two URLs:

`http://www.amazon.com:8080/gp/c/Moby_Dick`

`http://www.amazon.com/gp/c/Singularity_Sky?format=pdf&price=0`

but is not a match for any of these:

`https://www.amazon.com/gp/c/Hobbit`

`http://www.amazon.com /gp/c/Moby_Dick/`

`http://www.amazon.com/gp/c/Sex_and_the_City/Horror`

WebInspect treats elements of path segments matched by {...} groups in the rule URL as parameters, similar to those found in a query. Moreover, query parameters of crawled URLs matched by rule will be attacked along with parameters within the URL's path. In the following example of a matched URL, WebInspect would conduct attacks on the format and price parameters and on the third segment of the path (Singularity_Sky):

`http://www.amazon.com/gp/c/Singularity_Sky?format=pdf&price=0`

Asterisk Placeholder

The “*” placeholder may appear in the following productions and subproductions of the URL:

- Schema – as in `*://www.amazon.com/{param}`, which will match both HTTP and HTTPS.
- Authority – as in `http://*/{param}`, which will match all hosts, ports and userinfos.
 - Userinfo – as in `http://*@amazon.com/{param}`, which will match any username and password.
 - Username – as in `http://*:my_password@amazon.com/{param}`, which will match any username with given password.
 - Password – as in `http://john:*@amazon.com/{param}`, which will match any password for a given username.
- Hostname – as in `http://john:password@*/{param}`, which will match any host provided the username and password are as defined.
 - Host fragments – as in `http://*.amazon.com/{param}`, which will match any host within amazon.com domain.
 - Port – as in `http://www.amazon.com:*/{param}`, which will match any port for www.amazon.com host.
- Path – cannot be matched as a whole, since * in path matches a single segment or less.
 - Path segments – as in `http://www.amazon.com/gp/*/{param}`, which will match URLs with schema HTTP, hostname www.amazon.com, path containing three segments (first is exactly “gp”, second is any segment, and the third segment will be treated as parameter and won’t participate in matching).
 - Part of path segment – as in `http://www.amazon.com/gp/ref=*`, which will match URLs with schema HTTP, hostname www.amazon.com, path containing two segments (first is exactly “gp”, second containing any string with prefix “ref=”).
- Query – as in `http://www.amazon.com/gp/c/{param}?*`, which match any URL with schema HTTP, hostname www.amazon.com, path of three segments (first segment is “gp”, second segment is “c” and third segment being a parameter, so it won’t participate in matching); this URL also MUST contain a query string of arbitrary structure. Note the difference between rules `http://www.amazon.com/gp/c/{param}`and `http://www.amazon.com/gp/c/{param}?*`. First rule will match URL `http://www.amazon.com/gp/c/Three_Little_Blind_Mice`, while second will not.
 - Key-value pair of query – as in `http://www.amazon.com/gp/c/{param}?format=*` which will match URL only if query string has exactly one key-value pair, with key name being “format.”
 - Key-value pair of query – as in `http://www.amazon.com/gp/c/{param}?*=pdf` which will match URL only if query string has exactly one key-value pair, with value being “pdf.”
- Fragment – as in case `http://www.amazon.com/gp/c/{param}#*` which match any URL with fragment part being present

The main benefit of using placeholders is that it enables you to create rules that combine matrix parameters and URL path-based parameters within single rule. For relevant URL

```
http://www.amazon.com/gp/color;foreground=green;background=black/
something?format=dvi
```

the following rule will allow attacks on all parameters

```
gp/*/{param}
```

with the matrix parameter segment being ignored by * placeholder within second segment of the path, but recognized by WebInspect and attacked properly.

In the case of multiple rules matching a given URL, there are two options.

- Stop iterating over the rules once a match is found and so use only the first rule.
- Iterate over all of the rules and collect all custom parameters that match.

For instance, for the following URL

```
http://mySite.com/store/books/Areopagitica/32/1
```

the following rules both match

```
*/books/{booktitle}/32/{paragraph}
store/*/Areopagitica/{page}/{paragraph}
```

WebInspect will try to collect parameters from both rules to ensure the greatest attack coverage, so all three segments (“Areopagitica”, “32” and “1” in the example above) will be attacked.

Filters

Use these settings to add search-and-replace rules for HTTP requests and responses. This feature is used most often to avoid the disclosure of sensitive data such as credit card numbers, employee names, or social security numbers. It is a means of disguising information that you do not want to be viewed by persons who use WebInspect or those who have access to the raw data or generated reports.

If the text you specify is found, WebInspect reports it on the **Information** tab as a “Hidden Reference Found” vulnerability.

Filter HTTP Request Content

Use this area to specify search-and-replace rules for HTTP requests.

Filter HTTP Response Content

Use this area to specify search-and-replace rules for HTTP responses.

Follow the steps below to add a rule for finding or replacing keywords in requests or responses:

- 1 In either the **Request Content** or the **Response Content** group, click **Add**.
The *Add Request / Response Data Filter Criteria* window opens.
- 2 In the **Search For Text** box, type (or paste) the string you want to locate (or enter a regular expression representing the string).

Click  to insert regular expression notations or to launch the Regular Expression Editor (which facilitates the creation and testing of an expression).

- 3 In the **Search For Text In** box, select an area to search:
 - For Requests: select **All**, **Headers**, or **Postdata**.
 - For Responses: select **All**, **Headers**, or **Body** (that is, the code of the page itself)
- 4 Type (or paste) the replacement string in the **Replace search text with** box.
- 5 For case-sensitive searches, select the **Case-Sensitive Match** check box.
- 6 Click **OK**.

Cookies/Headers

Standard Header Parameters

[Include 'referer' in HTTP request headers](#)

Select this check box to include referer headers in WebInspect HTTP requests. The Referer request-header field allows the client to specify, for the server's benefit, the address (URI) of the resource from which the Request-URI was obtained.

[Include 'host' in HTTP request headers](#)

Select this check box to include host headers with WebInspect HTTP requests. The Host request-header field specifies the Internet host and port number of the resource being requested, as obtained from the original URI given by the user or referring resource (generally an HTTP URL).

Append Custom Headers

Use this section to add, edit, or delete headers that will be included with each audit WebInspect performs. For example, you could add a header such as "Alert: You are being attacked by Consultant ABC" that would be included with every request sent to your company's server when WebInspect is auditing that site. You can add multiple custom headers.

The default custom headers are described in the following table.

Header	Description
Accept: */*	Any encoding or file type is acceptable to the crawler.
Pragma: no-cache	This forces a fresh response; cached or proxied data is not acceptable.
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322)	Crawler is presenting itself as Microsoft Internet Explorer version 7.

Follow the steps below to add a custom header:

- 1 Click **Add**.

The *Specify Custom Header* window opens.

- 2 In the **Custom Header** box, enter the header using the format <name>: <value>.
- 3 Click **OK**.

Append Custom Cookies

Use this section to specify data that will be sent with the Cookie header in HTTP requests sent by WebInspect to the server when conducting a vulnerability scan.

Follow the steps below to add a custom cookie:

- 1 Click **Add**.

The *Specify Custom Cookie* window opens.

- 2 In the Custom Cookie box, enter the header using the format <name>=<value>.

For example, if you enter

`CustomCookie=ScanEngine`

then each HTTP-Request will contain the following header:

`Cookie: CustomCookie=ScanEngine`

- 3 Click **OK**.

Proxy

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 Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

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



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," or if the Internet Explorer setting "Use a proxy server for your LAN" is not selected, then a proxy server will not be used.

Configure proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering 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:

Authentication	Description
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. 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.
NT LAN Manager (NTLM)	NTLM is an authentication process used by all members of the Windows NT family of products. It 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.
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 Web Form Editor to determine the correct authentication type. Note that automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved.

Authentication	Description
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.

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 (described in the previous table).

Authentication

Authentication is the verification of identity as a security measure. Passwords and digital signatures are forms of authentication. You can configure automatic authentication so that a user name and password will be entered whenever WebInspect encounters a server or form that requires authentication. Otherwise, a crawl might be prematurely halted for lack of logon information.

Scan Requires Network Authentication

Select this check box if users must log on to your Web site or application using assigned credentials. You may then select the authentication method and specify the credentials.



WebInspect will crawl all servers granted access by this password. To avoid potential damage to your administrative systems, do not use a user name and password that has administrative rights. If you are unsure about your access rights, contact your System Administrator or internal security professional, or contact HP technical support.

See Authentication Types on [page 191](#) for a description of the available authentication types.

Client Certificates

Client certificate authentication allows users to present client certificates rather than entering a user name and password. Follow the steps below to use client certificates.

- 1 Select **Enable** in the **Client Certificates** group.
- 2 Click **Select** to open the *Client Certificates* window.
- 3 Choose a certificate.
- 4 Click **OK**.

Client Certificates for Tools

When using Step Mode or other tools that incorporate a proxy (specifically Web Macro Recorder, Web Proxy, Web Brute, and Web Form Editor), you may encounter servers that do not ask for a client certificate, even though a certificate is required. To accommodate this situation, you must edit the SPI.Net.Proxy.Config file using the following procedure:

Task 1: Find your certificate's serial number

- 1 Open Microsoft Internet Explorer.
- 2 Click **Tools** → **Internet Options**.
- 3 On the *Internet Options* window, select the **Content** tab and click **Certificates**.
- 4 On the *Certificates* window, select a certificate and click **View**.
- 5 On the *Certificate* window, click the **Details** tab.
- 6 Click the **Serial Number** field and copy the serial number that appears in the lower pane (highlight the number and press **Ctrl + C**).
- 7 Close all windows.

Task 2: Create an entry in the SPI.Net.Proxy.Config file

- 1 Open the SPI.Net.Proxy.Config file for editing. The default location is C:\Program Files\HP\HP WebInspect.
- 2 In the ClientCertificateOverrides section, add the following entry:

```
<ClientCertificateOverride HostRegex="RegularExpression"  
CertificateSerialNumber="Number"/>
```

where:

RegularExpression is a regular expression matching the host URL (example:
.austin\hp\com).

Number is the serial number obtained in Task 1.

- 3 Save the edited file.

Use a login macro for forms authentication

This type of macro is used primarily for Web form authentication. It incorporates logic that will prevent WebInspect from terminating prematurely if it inadvertently logs out of your application. When recording this type of macro, be sure to specify the application's log-out signature. Click the browse button  to locate and load a macro. To record a macro, enter the starting URL in the **Initial Recorder Location** box and click **Record**. The Web Macro Recorder then opens.

Login Macro Parameters

This section appears only if you have selected **Use a login macro for forms authentication** and the macro you have chosen or created contains fields that are designated as Smart Credentials (if you used the session-based or event-based Web Macro Recorder) or username and password parameters (if you used the TruClient Web Macro Recorder).

If you start a scan using a macro that includes Smart Credentials (or parameters for user name and password), then when you scan the page containing the input elements associated with these entries, WebInspect substitutes the user name and password specified here. This

allows you to create the macro using your own user name and password, yet when other persons run the scan using this macro, they can substitute their own user name and password.

Use a startup macro

This type of macro is used to acquire state by logging in to a particular area of the application, but does not contain logic that will prevent WebInspect from logging out. Use this type of macro if you cannot determine a logout signature or if the application cannot log you out.

Click the browse button  to locate the macro. Click **Record** to record a macro.

File Not Found

Determine ‘file not found’ (FNF) using HTTP response codes

Select this option to rely on HTTP response codes to detect a file-not-found response from the server.

- **Forced Valid Response Codes (Never an FNF)**—You can specify HTTP response codes that should never be treated as a file-not-found response.
- **Forced FNF Response Codes (Always an FNF)**—Specify those HTTP response codes that will always be treated as a file-not-found response. WebInspect will not process the response contents.

Enter a single response code or a range of response codes. For ranges, use a dash or hyphen to separate the first and last code in the list (for example, 400-404). You can specify multiple codes or ranges by separating each entry with a comma.

Determine ‘file not found’ from custom supplied signature

Use this area to add information about any custom 404 page notifications that your company uses. If your company has configured a different page to display when a 404 error occurs, add the information here. False positives can result in WebInspect from 404 pages that are unique to your site.

Auto detect ‘file not found’ page

Some Web sites do not return a status “404 Not Found” when a client requests a resource that does not exist. Instead, they may return a status “200 OK” but the response contains a message that the file cannot be found, or they might redirect to a home page or login page. Select this check box if you want WebInspect to detect these “custom” file-not-found pages.

WebInspect attempts to detect custom file-not-found pages by sending requests for resources that cannot possibly exist on the server. It then compares each response and measures the amount of text that differs between the responses. For example, most messages of this type have the same content (such as “Sorry, the page you requested was not found”), with the possible exception being the name of the requested resource.

If you select the **Auto detect** check box, you can specify what percentage of the response content must be the same. The default is 90 percent.

Policy

Select a policy to be used as the default whenever you start a scan.

You can substitute a different policy when starting a scan through the Scan Wizard, but the policy you select here will be used if you do not select an alternate.

You can also create, import, edit, or delete policies.

Create a Policy

Use the following procedure to create a policy:

- 1 Click **Create**.
The Policy Manager tool opens.
- 2 Select **File** → **New** (or click the New Policy icon).
- 3 Select the policy on which you will model a new one.
- 4 Refer to the on-line Help for additional instructions.

Import a Policy

Use the following procedure to import a policy:

- 1 Click **Import**.
- 2 On the *Import Custom Policy* window, click the browse button .
- 3 Using the **Files Of Type** list on the standard file-selection window, choose a policy type:
 - **Policy Files (*.policy)**—Policy files designed and created for versions of WebInspect beginning with release 7.0.
 - **Old Policy Files (*.apc)**—Policy files designed and created for versions of WebInspect prior to release 7.0.
 - **All Files (*.*)**—Files of any type, including non-policy files.
- 4 (optional) Edit the policy name.
- 5 Click **OK**.

A copy of the policy is created in the Policies folder (the default location is C:\Documents and Settings\All Users\Application Data\HP\HP WebInspect\Policies\). The policy and all of its enabled checks are imported into SecureBase using the specified policy name.



When importing policy files created for earlier versions of WebInspect, any custom check associated with that policy will be imported only if it can be found in the CustomAgents.xml file used by WebInspect 6.5 or earlier.

Delete a Policy

Use the following procedure to delete a policy:

- 1 Select a custom policy.
Only custom policies may be deleted.
- 2 Click **Delete**.

Edit a Policy

Use the following procedure to edit a policy:

- 1 Select a custom policy. Only custom policies may be edited.
- 2 Click **Edit**.

The Policy Manager tool opens. Refer to the on-line Help for additional instructions.

Crawl Settings

The WebInspect crawler is a software program designed to follow hyperlinks throughout a Web site, retrieving and indexing pages to document the hierarchical structure of the site. The parameters that control the manner in which WebInspect crawls a site are available in the Crawl Settings category.



These settings are not displayed if you select the **Audit only** option in the Scan Settings - Method category.

Link Parsing

WebInspect will follow all hyperlinks defined by HTML (using the `<a href>` tag) and those defined by scripts (if you select the JavaScript/VBScript analyzer option on the Crawl Settings - Content Analyzers panel). However, you may encounter other communications protocols that use a different syntax for specifying links. To accommodate this possibility, use the Custom Links feature to identify (using regular expressions) links that you want WebInspect to follow.

Follow the steps below to add a specialized link identifier:

- 1 Click **Add**.
The *Specialized Link Entry* window opens.
- 2 In the **Specialized Link Pattern** box, enter a regular expression designed to identify the link.
- 3 (Optional) Enter a description of the link in the **Comment** box.
- 4 Click **OK**.

Session Exclusions

All items specified in the Scan Settings - Session Exclusions are automatically replicated in the Session Exclusions for both the Crawl Settings and the Audit Settings. These items are listed in gray (not black) text. If you do not want these objects to be excluded from the crawl, you must remove them from the Scan Settings - Session Exclusions panel.

This panel (Crawl Settings - Session Exclusions) allows you to specify additional objects to be excluded from the crawl.

Excluded or Rejected File Extensions

If you select **Reject**, files having the specified extension will not be requested.

If you select **Exclude**, files having the specified extension will be requested, but will not be audited.

Follow the steps below to add a file extension:

- 1 Click **Add**.
The *Exclusion Extension* window opens.
- 2 In the **File Extension** box, enter a file extension.
- 3 Select either **Reject**, **Exclude**, or both.
- 4 Click **OK**.

Excluded MIME Types

Files associated with the MIME types you specify will not be audited.

Follow the steps below to add a MIME Type:

- 1 Click **Add**.
The *Provide a Mime-type to Exclude* window opens.
- 2 In the **Exclude Mime-type** box, enter a MIME type.
- 3 If you enter a regular expression to specify a MIME type, select the **Use Regular Expression** check box.
- 4 Click **OK**.

Other Exclusion/Rejection Criteria

You can identify various components of an HTTP message and then specify whether you want to exclude or reject a session (during the crawl) that contains that component.

- **Reject**—WebInspect will not send any HTTP requests to the host or URL you specify. For example, you should usually reject any URL that deals with logging off the site, since you don't want to log out of the application before the scan is completed.
- **Exclude**—During a crawl, WebInspect will not examine the specified URL or host for links to other resources. If you want to access the URL or host without processing the HTTP response, select the **Exclude** option, but do not select **Reject**. For example, to check for broken links on URLs that you don't want to process, select only the **Exclude** option.

To edit the default criteria:

- 1 Select a criterion and click **Edit** (on the right side of the **Other Exclusion/Rejection Criteria** list).
The *Reject or Exclude a Host or URL* window opens.
- 2 Select either **Host** or **URL**.
- 3 In the **Host/URL** box, enter a URL or fully qualified host name, or a regular expression designed to match the targeted URL or host.
- 4 Select either **Reject**, **Exclude**, or both.
- 5 Click **OK**.

To add exclusion/rejection criteria:

- 1 Click **Add** (on the right side of the **Excluded or Rejected URLs and Hosts** list).
The *Create Exclusion* window opens.
- 2 Select an item from the **Target** list.
- 3 If you selected **Query Parameter** or **Post Parameter** as the target, enter the **Target Name**.

- 4 From the **Match Type** list, select the method to be used for matching text in the target:
 - **Matches Regex:** Matches the regular expression you specify in the **Match String** box.
 - **Matches Regex Extension:** Matches a syntax available from HP's regular expression extensions you specify in the **Match String** box.
 - **Matches:** Matches the text string you specify in the **Match String** box.
 - **Contains:** Contains the text string you specify in the **Match String** box.
- 5 In the **Match String** box, enter the string or regular expression for which the target will be searched. Alternatively, if you selected a regular expression option in the **Match Type**, you can click the drop-down arrow and select **Create Regex** to launch the Regular Expression Editor.

- 6 Click .
- 7 (Optional) Repeat steps 2-6 to add more conditions. Multiple matches are ANDed.
- 8 If you are working in Current Settings, you can click **Test** to process the exclusions on the current scan. Any sessions from that scan that would have been filtered by the criteria will appear in the test screen, allowing you to modify your settings if required.
- 9 Click **OK**.
- 10 When the exclusion appears in the **Excluded or Rejected URLs or Hosts** list, select either **Reject**, **Exclude**, or both.

Example 1

To ensure that you ignore and never send requests to any resource at Microsoft.com, enter the following exclusion and select **Reject**.

Target	Target Name	Match Type	Match String
URL	n/a	contains	Microsoft.com

Example 2

Enter “logout” as the match string. If that string is found in any portion of the URL, the URL will be excluded or rejected (depending on which option you select). Using the “logout” example, WebInspect would exclude or reject URLs such as logout.asp or applogout.jsp.

Target	Target Name	Match Type	Match String
URL	n/a	contains	logout

Example 3

The following example rejects or excludes a session containing a query where the query parameter “username” equals “John.”.

Target	Target Name	Match Type	Match String
Query parameter	username	matches	John

Example 4

The following example excludes or rejects the following directories:

http://www.test.com/W3SVC55/
http://www.test.com/W3SVC5/
http://www.test.com/W3SVC550/.

Target	Target Name	Match Type	Match String
URL	n/a	matches regex	/W3SVC[0-9]*/

The default setting URL: \?[DNMSCO]=[ADNSM] is used for Apache directory indexing. These are sort options for the listing, which have no real impact on the page contents. An example would be http://www.w3.org/Icons/?C=M;O=A.

Audit Settings

An audit is the probe or attack conducted by WebInspect that is designed to detect vulnerabilities. The parameters that control the manner in which WebInspect conducts that probe are available from the Audit Settings category.

- These settings are not displayed if you select the **Crawl only** option or the **Manual crawl (step mode)** option in the Scan Settings - Method category.

Session Exclusions

All items specified in the Scan Settings - Session Exclusions are automatically replicated in the Session Exclusions for both the Crawl Settings and the Audit Settings. These items are listed in gray (not black) text. If you do not want these objects to be excluded from the audit, you must remove them from the Scan Settings - Session Exclusions panel.

This panel (Audit Settings - Session Exclusions) allows you to specify additional objects to be excluded from the audit.

Excluded or Rejected File Extensions

If you select **Reject**, files having the specified extension will not be requested.

If you select **Exclude**, files having the specified extension will be requested, but will not be audited.

Follow the steps below to add a file extension:

- 1 Click **Add**.

The *Exclusion Extension* window opens.

- 2 In the **File Extension** box, enter a file extension.
- 3 Select either **Reject**, **Exclude**, or both.
- 4 Click **OK**.

Excluded MIME Types

Files associated with the MIME types you specify will not be audited.

Follow the steps below to add a MIME type:

- 1 Click **Add**.

The *Provide a Mime-type to Exclude* window opens.

- 2 In the **Exclude Mime-type** box, enter a MIME type.
- 3 If you enter a regular expression to specify a MIME type, select the **Use Regular Expression** check box.
- 4 Click **OK**.

Other Exclusion/Rejection Criteria

You can identify various components of an HTTP message and then specify whether you want to exclude or reject a session (during the audit) that contains that component.

- **Reject**—WebInspect will not send any HTTP requests to the host or URL you specify. For example, you should usually reject any URL that deals with logging off the site, since you don't want to log out of the application before the scan is completed.
- **Exclude**—During the audit portion of the scan, WebInspect will not attack the specified host or URL. If you want to access the URL or host without processing the HTTP response, select the **Exclude** option, but do not select **Reject**. For example, to check for broken links on URLs that you don't want to process, select only the **Exclude** option.

To edit the default criteria:

- 1 Select a criterion and click **Edit** (on the right side of the **Other Exclusion/Rejection Criteria** list).
- The *Reject or Exclude a Host or URL* window opens.
- 2 Select either **Host** or **URL**.
 - 3 In the **Host/URL** box, enter a URL or fully qualified host name, or a regular expression designed to match the targeted URL or host.
 - 4 Select either **Reject**, **Exclude**, or both.
 - 5 Click **OK**.

To add exclusion/rejection criteria:

- 1 Click **Add** (on the right side of the **Excluded or Rejected URLs and Hosts** list).
- The *Create Exclusion* window opens.
- 2 Select an item from the **Target** list.
 - 3 If you selected **Query Parameter** or **Post Parameter** as the target, enter the **Target Name**.
 - 4 From the **Match Type** list, select the method to be used for matching text in the target:
 - **Matches Regex**: Matches the regular expression you specify in the **Match String** box.
 - **Matches Regex Extension**: Matches a syntax available from HP's regular expression extensions you specify in the **Match String** box.
 - **Matches**: Matches the text string you specify in the **Match String** box.
 - **Contains**: Contains the text string you specify in the **Match String** box.

- 5 In the **Match String** box, enter the string or regular expression for which the target will be searched. Alternatively, if you selected a regular expression option in the **Match Type**, you can click the drop-down arrow and select **Create Regex** to launch the Regular Expression Editor.

- 6 Click .

7 (Optional) Repeat steps 2-6 to add more conditions. Multiple matches are ANDed.

8 If you are working in Current Settings, you can click **Test** to process the exclusions on the current scan. Any sessions from that scan that would have been filtered by the criteria will appear in the test screen, allowing you to modify your settings if required.

9 Click **OK**.

- 10 When the exclusion appears in the **Excluded or Rejected URLs or Hosts** list, select either **Reject**, **Exclude**, or both.

Example 1

To ensure that you ignore and never send requests to any resource at Microsoft.com, enter the following exclusion and select **Reject**.

Target	Target Name	Match Type	Match String
URL	n/a	contains	Microsoft.com

Example 2

Enter “logout” as the match string. If that string is found in any portion of the URL, the URL will be excluded or rejected (depending on which option you select). Using the “logout” example, WebInspect would exclude or reject URLs such as logout.asp or applogout.jsp.

Target	Target Name	Match Type	Match String
URL	n/a	contains	logout

Example 3

The following example rejects or excludes a session containing a query where the query parameter “username” equals “John.”.

Target	Target Name	Match Type	Match String
Query parameter	username	matches	John

Example 4

The following example excludes or rejects the following directories:

<http://www.test.com/W3SVC55/>

<http://www.test.com/W3SVC5/>

<http://www.test.com/W3SVC550/>.

Target	Target Name	Match Type	Match String
URL	n/a	matches regex	/W3SVC[0-9]*/

Attack Exclusions

Excluded Parameters

Use this feature to prevent WebInspect from using certain parameters in the HTTP request to attack the Web site. This feature is used most often to avoid corrupting query and POSTDATA parameters.

To prevent certain parameters from being modified:

- 1 In the **Excluded Parameters** group, click **Add**.

The *Specify HTTP Exclusion* window opens.

- 2 In the **HTTP Parameter** box, enter the name of the parameter you want to exclude.

Click  to insert regular expression notations.

- 3 Choose the area in which the parameter may be found: **HTTP query data** or **HTTP POST data**. You can select both areas, if necessary.

- 4 Click **OK**.

Excluded Cookies

Use this feature to prevent WebInspect from using certain cookies in the HTTP request to attack the Web site. This feature is used to avoid corrupting cookie values.

This setting requires you to enter the name of a cookie. In the following example HTTP response ...

Set-Cookie: FirstCookie=Chocolate+Chip; path=/

...the name of the cookie is "FirstCookie."

Follow the steps below to exclude certain cookies.

- 1 In the **Excluded Cookies** group, click **Add**.

The Regular Expression Editor appears.

You can specify a cookie using either a text string or a regular expression.

- 2 To enter a text string:

- a In the **Expression** box, type a cookie name.

- b Click **OK**.

- 3 To enter a regular expression:

- a In the **Expression** box, type or paste a regular expression that you believe will match the text for which you are searching.

Click  to insert regular expression notations.

- b In the **Comparison Text** box, type or paste the text that is known to contain the string you want to find (as specified in the **Expression** box).
 - c To find only those occurrences matching the case of the expression, select the **Match Case** check box.
 - d If you want to replace the string identified by the regular expression, select the **Replace** check box and then type or select a string from the **Replace** box
 - e Click **Test** to search the comparison text for strings that match the regular expression. Matches will be highlighted in red.
 - f Did your regular expression identify the string?
- NO—Verify that the Comparison Text contains the string you want to identify or modify the regular expression.
- YES—Click **OK**.

Excluded Headers

Use this feature to prevent WebInspect from using certain headers in the HTTP request to attack the Web site. This feature is used to avoid corrupting header values.

To prevent certain headers from being modified, create a regular expression using the procedure described below.

- 1 In the **Excluded Headers** group, click **Add**.

The Regular Expression Editor appears.

You can specify a header using either a text string or a regular expression.

- 2 To enter a text string:

- a In the **Expression** box, type a header name.
- b Click **OK**.

- 3 To enter a regular expression:

- a In the **Expression** box, type or paste a regular expression that you believe will match the text for which you are searching.

- Click  to insert regular expression notations.
- b In the **Comparison Text** box, type or paste the text that is known to contain the string you want to find (as specified in the **Expression** box).
- c To find only those occurrences matching the case of the expression, select the **Match Case** check box.
- d If you want to replace the string identified by the regular expression, select the **Replace** check box and then type or select a string from the **Replace** box
- e Click **Test** to search the comparison text for strings that match the regular expression. Matches will be highlighted in red.
- f Did your regular expression identify the string?

NO—Verify that the Comparison Text contains the string you want to identify or modify the regular expression.

YES—Click **OK**.

Audit Inputs Editor

Using the Audit Inputs Editor, you can create additional parameters for audit engines and checks that require inputs.

To launch the tool, click **Audit Inputs Editor**.

To load inputs that you previously created using the editor, click **Import Audit Inputs**.

For detailed instructions on using the Audit Inputs Editor, see [Audit Inputs Editor](#) on page 236.

Attack Expressions

You may select one of the following language code-country code combinations (as used by the CultureInfo class in the .NET Framework Class Library):

- zh-cn: Chinese and China
- ja-jp: Japanese and Japan
- ko-Kr: Korean and Korea

The CultureInfo class holds culture-specific information, such as the associated language, sublanguage, country/region, calendar, and cultural conventions. This class also provides access to culture-specific instances of DateTimeFormatInfo, NumberFormatInfo, CompareInfo, and TextInfo. These objects contain the information required for culture-specific operations, such as casing, formatting dates and numbers, and comparing strings.

Vulnerability Filtering

By applying certain filters (listed below), you can modify the results of a vulnerability scan to accommodate your specific testing environment.

- **Standard Vulnerability Definition:** This filter reports vulnerabilities in the same manner as QAInspect.
- **403 Blocker:** This filter revokes vulnerabilities when the status code of the vulnerable session is 403 (Forbidden).
- **Parameter Vulnerability Roll-Up:** This filter consolidates multiple parameter manipulation and parameter injection vulnerabilities discovered during a single session into one vulnerability.
- **Response Inspection DOM Event Parent-Child:** This filter disregards a keyword search vulnerability found in JavaScript if the same vulnerability has already been detected in the parent session.

Adding a Filter

To add a filter to your default settings:

- 1 Click **Edit → Default Scan Settings**.
- 2 In the **Audit Settings** panel in the left column, select **Vulnerability Filtering**.

All available filters are listed in either the **Disabled Filters** list or the **Enabled Filters** list.

- 3 To enable a filter, select a filter in the **Disabled Filters** list and click **Add**.

The filter is removed from the Disabled Filters list and added to the Enabled Filters list.

- 4 To disable a filter, select a filter in the **Enabled Filters** area and click **Remove**.
The filter is removed from the **Enabled Filters** list and added to the **Disabled Filters** list.

Smart Scan

Smart Scan is an “intelligent” feature that discovers the type of server that is hosting the Web site and checks for known vulnerabilities against that specific server type. For example, if you are scanning a site hosted on an IIS server, WebInspect will probe only for those vulnerabilities to which IIS is susceptible. It would not check for vulnerabilities that affect other servers, such as Apache or iPlanet.

If you select **Enable Smart Scan**, you can choose one or more of the identification methods described below.

Use regular expressions on HTTP responses

This method, employed by previous releases of WebInspect, searches the server response for strings that match predefined regular expressions designed to identify specific servers.

Use server analyzer fingerprinting and request sampling

This advanced method sends a series of HTTP requests and then analyzes the responses to determine the server/application type.

Custom server/application type definitions

If you know the server type for a target domain, you can select it using the **Custom server/application type definitions** section. This identification method overrides any other selected method for the server you specify.

- 1 Click **Add**.

The *Server/Application Type Entry* window opens.

- 2 In the **Host** box, enter the domain name or host, or the server’s IP address.
- 3 (Optional) Click **Identify**.

WebInspect contacts the server and uses the server analyzer fingerprinting method to determine the server type. If successful, it selects the corresponding check box in the **Server/Application Type** list.

Alternatively, if you select the **Use Regular Expressions** option, enter a regular expression designed to identify a server. Click  to insert regular expression notations or to launch the Regular Expression Editor (which facilitates the creation and testing of an expression).

- 4 Select one or more entries from the **Server/Application Type** list.
- 5 Click **OK**.

Importing and Exporting Settings

If you require different settings for different scan actions, you can save your settings in an XML-formatted file and load them when needed. You can also reload the WebInspect factory default settings.

- 1 Click **Edit → Default Settings**.

The *Default Settings* window appears.

- 2 To export settings:

- a Click **Save settings as** (at the bottom of the left pane).
 - b On the *Save Scan Settings* window, select a folder and enter a file name.
 - c Click **Save**.

- 3 To import settings:

- a Click **Load settings from file** (at the bottom of the left pane).
 - b On the *Open Scan Settings File* window, select a file.
 - c Click **Open**.

- 4 To restore factory default settings:

- a Click **Restore factory defaults** (at the bottom of the left pane).
 - b When prompted to confirm your selection, click **Yes**.

Managing Settings

Creating a Settings File

Follow the steps below to create a new settings file based on the default settings:

- 1 On the WebInspect menu bar, click **Edit → Manage Settings**.
- 2 On the *Manage Settings* window, click **Add**.

The *Create New Settings* window opens.

- 3 Change settings.
- 4 When finished, click **OK**.
- 5 On the *Save Scan Settings File* window, enter a file name and location.
- 6 Click **Save**.

Editing or Deleting a Settings File

Follow the steps below to edit or delete settings from WebInspect:

- 1 On the WebInspect menu bar, click **Edit → Manage Settings**.
- 2 On the *Manage Settings* window:
 - a To delete a settings file, select a file and click **Delete**.

- b To edit a settings files, select a file and click **Edit**.

Exporting a Settings File

Follow the steps below to export a settings from WebInspect:

- 1 Select a file.
- 2 Click **Export**.
- 3 Using a standard file-selection dialog, name the file and select a location.
- 4 Click **Save**.

Transferring Settings to or from an Enterprise Server

Use this feature to:

- Create an AMP or WebInspect Enterprise scan template based on a WebInspect settings file and upload it from WebInspect to the enterprise server.
- Create a WebInspect settings file based on an AMP or WebInspect Enterprise scan template and download it from the enterprise server to WebInspect.

WebInspect settings files and AMP/WebInspect Enterprise scan templates do not have the same format; not all settings in one format are replicated in the other. Note the warnings that follow descriptions of the conversion procedure.

To create an AMP/WebInspect Enterprise scan template:

- 1 Click the WebInspect **Enterprise Server** menu and select **Connect to AMP** or **Connect to WebInspect Enterprise**
- 2 Click the **Enterprise Server** → **Transfer Settings**.
- 3 On the *Settings Transfer* window, select a WebInspect settings file from the **Local Settings File** list.
- 4 (Optional) Click **View** to review the settings as they appear in a WebInspect settings file. To continue, click **Close**.

This is a read-only file. Any changes you make will not be persisted.

- 5 For WebInspect Enterprise, select the Project and Project Version to which the template will be transferred.
- 6 If necessary, click **Refresh** to ensure the lists include the latest settings files and scan templates.
- 7 For WebInspect Enterprise, enter the name of the scan template that will be created. You cannot duplicate the name of an existing template.
- 8 Click **Upload**.
- 9 For AMP, when prompted to select a name, enter a name for the scan template that will be created. You cannot duplicate the name of an existing template. Click **OK**.

10 Click OK.

- 
 - All template settings that are not extracted from WebInspect will use the AMP or WebInspect Enterprise template default settings.
 - The scan template will not specify the policy used by the WebInspect settings file. Instead, it will contain the “Use Any” option.
 - Any client certificate information that may be included in the WebInspect settings file is transferred to the scan template, but the certificates are not transmitted.
 - All WebInspect settings are preserved in the scan template, even if they are not used by AMP or WebInspect Enterprise. Therefore, if you subsequently create a WebInspect settings file based on the enterprise scan template you created from the original settings file, the WebInspect settings will be retained.

To create a WebInspect settings file:

- 1 Click the WebInspect **Enterprise Server** menu and select **Transfer Settings**.
- 2 For WebInspect Enterprise, select the Project and Project Version from which the template will be transferred.
- 3 On the *Settings Transfer* window, select a scan template from the list.
- 4 (Optional) Click **View** to review the settings as they would appear in a WebInspect settings file. To continue, click **Close**.

This is a read-only file. Any changes you make will not be persisted.

- 5 If necessary, click **Refresh** to ensure the lists include the latest settings files and scan templates.
- 6 Click **Download**.
- 7 Using a standard file-selection window, name the settings file, select a location in which to save it, and click **Save**.

 The WebInspect settings file will not specify the policy used by the enterprise scan template. Instead, it will specify the Standard policy.

6 Application Settings

Introduction

To access this feature, click **Edit → Application Settings**.

The Application Settings are divided into the following categories:

- General
- Directories
- Server Profiler
- Logging
- Reports
- Smart Update
- HP Quality Center
- Database
- License
- Step Mode
- Proxy
- Run as a Service
- Support Channel
- IBM Rational ClearQuest

General

General Settings

Enable Active Content in Browser Views

Select this option to allow execution of JavaScript and other dynamic content in all browser windows within WebInspect.

For example, one WebInspect attack tests for cross-site scripting by attempting to embed a script in a dynamically generated Web page. That script instructs the server to display an alert containing the number “76712.” If active content is enabled and if the attack is successful (i.e., cross-site scripting is possible), then selecting the vulnerable session and clicking on **Web Browser** in the Session Info panel will execute the script and display the following:



If you initiate or open a scan while this option is not enabled, and you then enable this option, the browser will not execute the active content until you close and then reopen the scan.

Enable Diagnostic File Creation

If the WebInspect application should ever fail, this option forces WebInspect to create a file containing data that was stored in main memory at the time of failure. The file can be transferred to HP support personnel using the HP Support Channel Tool.

If you select this option, you may also specify how many diagnostic files should be retained. When the number of files exceeds this limit, the oldest file will be deleted.

Reset “Don't Show Me Again” messages

By default, WebInspect displays various prompts and dialogs to remind you of certain consequences that may occur as a result of an action you take. These dialogs contain a check box labeled “Don't show me again.” If you select that option, WebInspect discontinues displaying those messages. You can force WebInspect to resume displaying those messages if you click **Reset “Don't Show Me Again” messages**.

HP SecurityScope Settings

Use SecurityScope information when encountered on target site

If this option is selected and WebInspect detects that HP SecurityScope is installed on a target server, it will incorporate SecurityScope information to improve overall scan efficiency.

A notation on the WebInspect dashboard indicates whether or not SecurityScope has been detected.

Automatically group by duplicate vulnerabilities in Vulnerability window

If this option is selected and HP SecurityScope information is used (above setting), then vulnerabilities listed on the **Vulnerability** tab in the Summary pane will be grouped by check and then by equivalent vulnerabilities.

Macro Recorder

WebInspect accommodates three different macro recorders. Select the one that will be launched by default when creating a macro. The choices are:

- Traffic-Mode: Session-based; can be used with either Internet Explorer or Mozilla Firefox; has limited capacity for Silverlight and Flash.
- Event-Based (IE Compatible): Does not support the recording of Flash or Silverlight applications; more reliable than traffic-mode recorder; can be used with either Internet Explorer or Mozilla Firefox.
- Event-Based (TruClient): Easy-to-use adaptation of the Ajax TruClient technology originally developed for use with HP LoadRunner and HP Performance Center; does not support the recording of Flash or Silverlight applications; uses Mozilla Firefox (embedded); cannot replicate a challenge/response test where the challenge varies each time.

Note: When recording a macro for a workflow-driven scan, WebInspect always uses the traffic-mode recorder.

Database

Connection Settings for Scan/Report Storage

You can store WebInspect scan and report data in a database created using either Microsoft SQL Server Standard Edition or Microsoft SQL Server Express Edition.

- **Use SQL Server Express**—Select this option to save scan data in Microsoft SQL Server Express Edition. Data for each scan is stored in a separate database. The maximum size is 4 GB (unless you are using SQL Server 2008 R2 Express, which has a maximum database storage of 10GB).
- **Use SQL Server**—Select this option to save scan data in Microsoft SQL Server Standard Edition. You can configure multiple database settings and assign a “profile name” to each collection of settings, allowing you to switch easily from one configuration to another.

To configure a profile for SQL Server Standard Edition:

- 1 Click **Configure** (to the right of the drop-down list).
- 2 On the *Manage Database Settings* window, click **Add**.
- 3 Enter a name for this database profile.
- 4 Select a server from the **Server Name** list.
- 5 In the **Log on to the server** group, specify the type of authentication used for the selected server:
 - **Use Windows Authentication**—Log on by submitting the user's Windows account name and password.
 - **Use SQL Server Authentication**—Use SQL Server authentication, which relies on the internal user list maintained by the SQL Server computer. Enter the user name and password.
- 6 Enter or select a specific database, or click **New** to create a database.
- 7 Click **OK** to close the *Add Database* window.
- 8 Click **OK** to close the *Manage Database Settings* window.

Connection Settings for Scan Viewing

When displaying a list of scans (using either the Manage Scans view or the Report Generator wizard), WebInspect can access scan data stored in SQL Server Standard Edition and/or SQL Server Express Edition. You can select either or both options.

- **Show Scans Stored in SQL Server Express**—Select this option if you want to access scan data stored in a local SQL Server Express Edition.
- **Show Scans Stored in SQL Server**—Select this option if you want to access data in SQL Server Standard Edition. See Use SQL Server Standard (above) for instructions.

Directories

You can specify the default directories in which WebInspect elements are stored, using the following procedure:

- 1 Click the browse button  next to a category of information.
- 2 Use the *Browse For Folder* window to select or create a directory.
- 3 Click **OK**.

License

License Details

This section provides pertinent information about the WebInspect license.

If you want to change certain provisions of the license, click **Configure Licensing**, which will invoke the HP License Wizard.

The contents of the lower section of the window depend on the type of license management currently employed:

- Connected directly to the HP license server
- Connected to a local License and Infrastructure Manager (LIM).

Direct Connection to HP

Update

If you upgrade from a trial version or if you otherwise modify the conditions of your license, click **Update**. The application will contact the license server and update the information stored locally on your machine.

Deactivate

WebInspect licenses are assigned to specific computers. If you would like to transfer this license to a different computer:

- 1 Copy the activation token.

Take care not to lose or misplace this number. Write it or print it, and keep it in a safe place.

- 2 Click **Deactivate**.

The application contacts the license server and releases your license, allowing you to install WebInspect on another computer.

- 3 At the new computer, access the WebInspect application settings for licensing and enter the activation token.

Connection to LIM

Select the manner in which you want the License and Infrastructure Manager to handle the WebInspect license assigned to this computer.

Connected License

The computer can run the HP product only when the computer is able to contact the LIM. Each time you start the HP software, the LIM allocates a seat from the license pool to this installation. When you close the software, the seat is released from the computer and allocated back to the pool, allowing another user to consume the license.

Detached License

The computer can run the HP product anywhere, even when disconnected from your corporate intranet (on which the LIM is normally located), but only until the expiration date you specify. This allows you to take your laptop to a remote site and run the HP software. When you reconnect to the corporate intranet, you can access the Application License settings and reconfigure from Detached to Connected.

Server Profiler

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

To enable this preliminary examination, select **Run Profiler Automatically** on Step 4.

By default, 10 specific tests are enabled. To exclude a test, clear its associated check box.

Check for case-sensitive servers

This module determines if the host server is case-sensitive when discriminating among URLs. For example, some servers (such as IIS) do not differentiate between www.mycompany.com/samplepage.htm and www.mycompany.com/SamplePage.htm. If the profiler determines that the server is not case-sensitive, you can disable WebInspect's case-sensitive feature, which would improve the speed and accuracy of the crawl.

Check 'Maximum Folder Depth' setting

The maximum folder depth setting is intended primarily for sites that programmatically append subfolders to URLs. Without such a limit, WebInspect would endlessly crawl these dynamic folders. This module determines if the site contains valid URLs that extend beyond that limit and, if so, allows you to increase the setting.

Verify client authentication protocol

This module determines which authentication (sign-in) protocol, if any, is required. WebInspect supports HTTP Basic, NTLM, Digest, Kerberos.

[Check for additional hosts](#)

This module searches the target site for references to additional host servers and allows you to include them as allowed hosts.

[Reveal navigation parameters](#)

This module determines if the target site uses query parameters in URLs to specify the content of the page and, if so, displays a list of parameters and values that were encountered during the analysis. You can select one or more parameters for WebInspect to use during the scan.

[Check for non-standard ‘file not found’ responses](#)

This module determines if a site returns a response code other than 404 when the client requests a non-existent resource. Recognizing this will prevent WebInspect from auditing non-essential responses. During a scan, WebInspect records how many attacks were conducted with the “file not found” detection module enabled and how many of those attacks were scored as “file not found.” If fewer than 90 percent of these attacks are scored as real files instead of “file not found,” then thyself not found” module may not be working correctly. Also, if fewer than 100 attacks were sent, the module will not perform the analysis and will not return any recommendations.

[Check for session state embedded in URLs](#)

Instead of using cookies, some servers embed session state in URLs. WebInspect detects this practice by analyzing the URL with regular expressions. This module attempts to determine if changes to the regular expressions are required.

[Analyze thread count](#)

This module determines if the thread count should be lowered. Relatively high thread counts, while enabling a faster scan, can sometimes exhaust server resources.

[Check for invalid audit exclusions](#)

WebInspect settings prevent pages with certain file extensions from being audited ([Session Exclusions](#) on page 199). The specified extensions are for pages that ordinarily do not have query parameters in the URL of the request. If the settings are incorrect, the audit will not be as thorough. The profiler can detect when pages having audit-excluded extensions actually contain query parameters and will recommend removing those exclusions.

[Verify maximum response size](#)

A WebInspect scan setting specifies the maximum response size allowed; the default is 1,000 kilobytes. This module attempts to detect responses larger than the maximum and, if found, recommends that you increase the limit.

[Optimize settings for specific applications](#)

This module determines if you are scanning a well-known test site (such as WebGoat, Hacme Bank, etc.) and determines if WebInspect has a prepopulated settings file (a template) designed specifically for that site. These templates are configured to optimize the crawl, audit, and performance of your scans.

Add/Remove Trailing Slash

This module determines if the target site requires or prohibits a trailing slash on the start URL.

Step Mode

Select an option from the **Default Audit Mode** list. The choices are:

- **Audit as you browse**—While you are navigating a target Web site, WebInspect concurrently audits the pages you visit.
- **Manual Audit**—This option allows you to pause the Step Mode scan and return to WebInspect, where you can select a specific session and audit it.

Step Mode requires a proxy. Specify the IP address that the proxy should use and then specify the port (or select **Automatically Assign Port**).

Logging

Clear Logs

Click this button to clear all logs.

Minimum Logging Level

Specify how WebInspect should log different functions and events that occur within the application. The choices are (from most verbose to least verbose) **Debug**, **Info**, **Warn**, **Error**, and **Fatal**.

Threshold for Log Purging

If you do not select **Never Purge**, WebInspect deletes all logs when either the total amount of disk space used by all logs exceeds the size you specify or the number of logs exceeds the number you specify.

Rolling Log File Maximum Size

Specify the maximum size (in kilobytes) that any log file may attain. When a file reaches this limit, WebInspect will simply stop writing to it.

Proxy

WebInspect Web services are used for update and support communications.

If you are not using a proxy server to access these services, select **Direct Connection (proxy disabled)**.

If you are required to use a proxy server to access these services, select one of the following.

[Auto detect proxy settings](#)

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and use this to configure the browser's Web proxy settings.

[Use Internet Explorer proxy settings](#)

Import your proxy server information from Internet Explorer.

[Use Firefox proxy settings](#)

Import your 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," or if the Internet Explorer setting "Use a proxy server for your LAN" is not selected, then a proxy will not be used. To access browser proxy settings:

Internet Explorer: **Tools** → **Internet Options** → **Connections** → **LAN Settings**

Firefox: **Tools** → **Options** → **Advanced** → **Network** → **Settings**

[Configure a proxy using a PAC file](#)

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

[Explicitly configure proxy](#)

Configure a proxy by entering 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 From the **Type** list, 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.
- 4 If your proxy server requires authentication, enter the qualifying user name and password.

Reports

Options

[Always prompt to save favorites](#)

A "favorite" is simply a named collection of one or more reports and their associated parameters. When using the Report Generator, you can select reports and parameters, and then select **Favorites** → **Add to favorites** to create the combination. If you select this option, then WebInspect will prompt you to save the favorite whenever you modify it by adding or removing a report.

Smart truncate vulnerability text

Generated reports can contain very lengthy HTTP request and response messages. To save space and help focus on the pertinent data related to a vulnerability, you can exclude message content that precedes and follows the data that identifies or confirms the vulnerability (identified by red highlighting).

The following example illustrates the report of a cross-site scripting vulnerability using “smart” truncation and a padding size of 20 characters. The complete header is always reported. The remaining message text is deleted, except for the vulnerability and the 20 characters preceding it and the 20 characters following it. The retained text is then bracketed by the notation “...TRUNCATED...” to indicate that truncation has occurred. Note that the length of the original message was 2,377 characters (Content-Length: 2377).

Response:

```
HTTP/1.1 200 OK
Date: Tue, 04 Aug 2009 17:35:10 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
Content-Length: 2377
Content-Type: text/html
Cache-control: private

...TRUNCATED...1>Household Checking<script>alert(53316)</script></td>
</tr>

<tr>...TRUNCATED...
```

To use smart truncation in reports, select **Smart truncate vulnerability text** and then specify the number of characters to retain preceding and following the data that identifies or confirms the vulnerability. A maximum of 10 vulnerabilities can be reported in a single request or response.



Note: This feature functions as described only if the report controls containing the RequestText and ResponseText data fields have the TruncateVulnerability property set to True and the MaxLength property set to zero. If TruncateVulnerability is set to True and the MaxLength property is nonzero, then the application setting for padding size is overridden by the MaxLength value.

Headers and Footers

Select a template containing the headers and footers to be used by default on all reports. Also, if necessary, enter the requested parameters.

The WebInspect Master Report used three images to create a report.

- The cover page image appears in the center of the cover page, with the top of the image approximately 3.5 inches from the top.
- The header logo image appears on the left side of the header on every page.

Run as a Service

Interactive

This configuration information is used for integrating WebInspect into the Assessment Management Platform (AMP) or WebInspect Enterprise as an interactive scanning device. This means that you should initiate scans through the WebInspect graphical user interface rather than using the AMP Web console or the WebInspect Enterprise Web console. Scan results can then be uploaded to and downloaded from the AMP or WebInspect Enterprise system. However, those systems still controls which IP addresses may be scanned and also provides the SecureBase data and scanning policies to WebInspect.

Enter the URL or IP address of the manager.

If you select **Auto Upload Scans**, WebInspect automatically uploads scans to the manager upon completion of the scan.

Sensor

This configuration information is used for integrating WebInspect into the Assessment Management Platform (AMP) or WebInspect Enterprise as a sensor. After providing the information and starting the sensor service, you should use AMP console or WebInspect Enterprise console, not the WebInspect graphical user interface, to conduct scans.

Manager URL

Enter the URL or IP address of the Manager.

Sensor Authentication

Enter a user name (formatted as domain\username) and password, then click **Test** to verify the entry.

Enable Proxy

If WebInspect must go through a proxy server to reach the AMP manager or WebInspect Enterprise manager, select **Enable Proxy** and then provide the IP address and port number of the server. If authentication is required, enter a valid user name and password.

Override Database Settings

WebInspect normally stores scan data in the device you specify in the Application Settings (**Edit** → **Application Settings** → **Database**). However, if WebInspect is connected to AMP or WebInspect Enterprise as a sensor, you can select this option and then click **Configure** to specify an alternative device.

Service Account

Select one of the following options to specify the account under which the service should run:

- **Local system account**—The LocalSystem account is a predefined local account used by the service control manager. The service has complete unrestricted access to local resources.
- **This account**—Identify the account and provide the password.

Sensor Status

This area displays the current status of the Sensor Service and provides buttons allowing you to start or stop the service.

After configuring WebInspect as a sensor, click **Start**.

Smart Update

- 1 In the **Service** box, enter the URL for the Smart Update service. The default is:
`https://smartupdate.hpsmartupdate.com/`
- 2 Select **Enable Smart Update on Startup** if you want to check for updates automatically when starting WebInspect.

Support Channel

The HP support channel allows WebInspect to send data to and download messages from HP. It is used primarily for sending logs and “false positive” reports and for receiving “What’s New” notices.

Select **Allow connection to Hewlett-Packard** to open the HP support channel. You may then specify the following:

- Support Channel URL. The default is:
`https://SupportChannel.HPSmartUpdate.com/SupportChannel/service.asmx`
- Upload Directory. The default is:
`C:\Documents and Settings\All Users\Application Data\HP\HP WebInspect\SupportChannel\Upload\`
- Download Directory. The default is:
`C:\Documents and Settings\All Users\Application Data\HP\HP WebInspect\SupportChannel\Download\`

HP Quality Center

To integrate WebInspect with HP Quality Center, you must create one or more profiles that describe the Quality Center server, project, defect priority, and other attributes. You can then convert a WebInspect vulnerability to a Quality Center defect and add it to the Quality Center database.

If you intend to run HP Quality Center and WebInspect on the same machine, then before creating a profile, you must download the HP Quality Center client application. To do so, simply open your Web browser and enter the Quality Center URL in the Address bar; then click the Quality Center link and log in. This prerequisite does not apply if you are connecting to Quality Center version 11, also known as Application Lifecycle Management (ALM).

Follow the steps below to add a profile.

- 1 Click **Add**.
 - 2 On the Add *Profile* window, enter a profile name.
 - 3 The **Server URL** box is populated (if possible) with the URL of the Quality Center server. If you enter or modify the URL, use the format `http://<qc-server>/qcbin/`. Do not append “start_a.htm” (or other file name) to the URL.
 - 4 Enter the user name and password that will allow you to access the server.
 - 5 Click **Authenticate**.
- If the authentication credentials are accepted, the server populates the **Domain** and **Project** lists.
- 6 Click **Connect**.
 - 7 In the **Defect Reporting** group, select a subject.
 - 8 From the **Defect priority** list, select a priority that will be assigned to all WebInspect vulnerabilities reported to Quality Center using this profile.
 - 9 Use the **Assign defects to** list to select the person to whom the defect will be assigned.
 - 10 Select an entry from the **Project found in** list.
 - 11 Use the remaining lists to map the WebInspect vulnerability rating to an HP Quality Center defect rating. If you select **Do Not Publish**, the vulnerability will not be exported. You must select at least one of the file mappings.
 - 12 To export notes and screenshots associated with a WebInspect vulnerability, select **Upload vulnerability attachments to defect**.
 - 13 In the **Required/Optional Fields** group, double-click an entry and enter or select the requested information. If you try to save your work without supplying a required field, WebInspect prompts you to enter it.

Note: Creating or editing a profile consumes a license issued to Quality Center. The license is released, however, when the HP Quality Center application settings are closed. Similarly, sending a vulnerability to Quality Center consumes a license, but it is released after the vulnerability is sent.

IBM Rational ClearQuest

To integrate WebInspect with IBM Rational ClearQuest, you must create one or more profiles that describe the server, project, defect priority, and other attributes. You can then convert a WebInspect vulnerability to an IBM Rational ClearQuest defect and add it to the Rational ClearQuest database.

Follow the steps below to add a profile. IBM Rational ClearQuest must be installed before you can complete this form.

- 1 Click **Add**.
- 2 Enter a profile name and click **OK**.
The configuration settings become enabled.
- 3 Select a database set from the list (or accept the default).
- 4 Select a database from the **Database** list (or accept the default).

- 5 Enter your credentials in the **User Name** and **Password** boxes.
 - 6 Click **Authenticate**.
- If successful, the **Defect Reporting** section becomes enabled.
- 7 Complete the fields in the **Defect Reporting** section. You must select at least one of the five “map to” categories.
 - a In the **Headline Prefix** box, enter a text string that will be prepended to the title of each WebInspect vulnerability.
 - b From the **Owner** list, select the person to whom the defect will be assigned.
 - c Select a project from the **Project** list.
 - d From the **Priority** list, select a priority that will be assigned to all WebInspect vulnerabilities reported to Rational ClearQuest using this profile.
 - e To enable the upload of attachments associated with vulnerabilities, select **Upload vulnerability attachments to defect**.
 - f WebInspect ranks vulnerabilities as either critical, high, medium, low, or informational. Use the **Map <severity> Risks to** lists to specify how these ratings should be equated to those used by Rational ClearQuest. The default selection is “Do Not Publish.”
 - g Edit the **Mandatory/Optional** fields. Click an entry in the **Value** column to open the appropriate editor. Any field that contains “Mandatory” in the **Requiredness** column must have a value specified. If you try to save your work without supplying a mandatory field, WebInspect prompts you to enter it.

For lists that are not limited, the editor displays a checked list box; you can add items to the list by checking **Add New Item** (at the bottom). For the Date/Time editor, both date and time are displayed, regardless of the actual IBM Rational ClearQuest usage (which could be date, time, or date/time).

7 WebInspect Tools

Introduction

A robust set of diagnostic and penetration testing tools is packaged with WebInspect. These include:

- Audit Inputs Editor
- Compliance Manager
- Cookie Cruncher
- Encoders/Decoders
- HTTP Editor
- Log Viewer
- Policy Manager
- Regular Expression Editor
- Report Designer
- Server Analyzer
- Server Profiler
- Smart Update
- SQL Injector
- SWFScan
- Web Brute
- Web Discovery
- Web Form Editor
- Web Fuzzer
- Web Macro Recorder (Traffic-Mode)
- Web Macro Recorder (Event-Based)
- Web Macro Recorder (TruClient)
- Web Proxy
- Web Service Test Designer
- Web Application Firewall Integration Tool

WebInspect installations also include the HP Support tool, which provides a quick and simple method for uploading files that may help HP support personnel analyze and resolve any problems you encounter while using Application Security Center products.

Client Certificates

When using tools that incorporate a proxy, you may encounter servers that do not ask for a client certificate, even though a certificate is required. To accommodate this situation, you must edit the SPI.Net.Proxy.Config file. For details, see [Client Certificates for Tools](#) on page 179.

Policy Manager

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

- Audit Engines
- General Application Testing
- General Text Searching
- Third-Party Web Applications
- Web Frameworks/Languages
- Web Servers
- Web Site Discovery
- 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 Web site for vulnerabilities.

WebInspect contains several prepackaged policies designed to accommodate the majority of 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

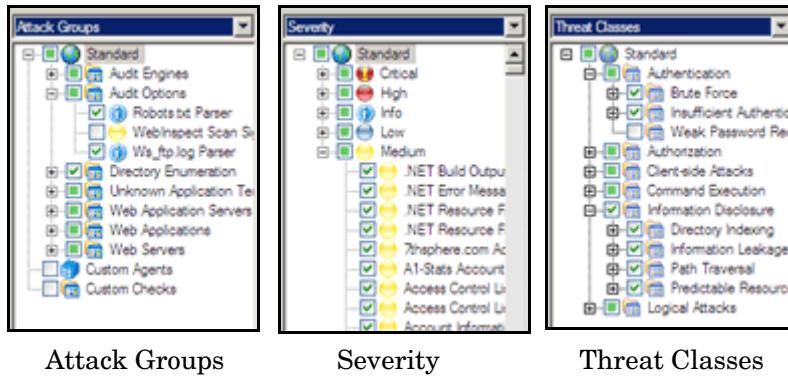
The Policy Manager has two different views, selectable from the **View** menu or by clicking icons on the toolbar. They are:

- Standard
- Search

Standard View

This view displays, by default, a list of checks categorized by threat class (according to classifications established by the Web Application Security Consortium). Alternatively, a drop-down list allows you to display all attack agents by severity, or a list of audit engines and attack groups.

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



Attack Groups

Severity

Threat Classes

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

- 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.

Search View

The Search view allows you to locate attack agents containing the text you specify in a selected report field (i.e., summary, implication, execution, recommendation, and fix).

A screenshot of the Policy Manager application window titled "Policy Manager". The window has a menu bar with File, Edit, View, Tools, Help. Below the menu is a toolbar with icons for Standard View and Search View, where "Search View" is highlighted. The main area is titled "Criteria:" with a dropdown set to "CWE ID" and a search input field containing "7". A "Search" button is to the right. Below this is a table titled "Vulnerabilities (1115)" with columns: Name, ID, Type, Severity, and Last Updated. The table lists several vulnerabilities, including "Counter.exe Web Hit Counter DOS Attack" (ID 36, ABS, High, last updated 10/14/2009 3:41:05 PM), "Apache Access Control List Disclosure (.htaccess)" (ID 38, COMMONFILE, Medium, last updated 3/11/2009 2:17:22 PM), "IIS Frontpage Server Extensions Author.dll Possib..." (ID 77, ABS, Critical, last updated 9/15/2009 6:09:40 PM), "Frontpage Server Extensions Shtml.dll Multiple Po..." (ID 91, ABS, High, last updated 8/13/2007 2:55:38 PM), "IIS ism.dll Multiple Possible Vulnerabilities" (ID 144, ABS, Critical, last updated 11/22/2005 3:36:56 PM), "WebLog Administrative Access Bypass" (ID 365, ABS, High, last updated 7/17/2006 2:25:33 PM), "Siteminder Administration Interface" (ID 383, ABS, Medium, last updated 11/22/2005 3:36:56 PM), and "Alibaba Web Server Arbitrary Command Executio..." (ID 494, ABS, Critical, last updated 11/22/2005 3:36:56 PM). At the bottom left, it says "Attack Group: Third-Party Web Applications/Statistics/Logging" and "Summary". The summary pane shows the details for the selected vulnerability: "Summary: Counter.exe Web Hit Counter DOS Attack", "Vulnerability ID: 36", "CWE ID: 730_400", and a description: "A set of vulnerabilities in the counter.exe web hit counter program enables denial of service attacks." At the bottom, there is a "Find Vulnerability in Standard View" link and a status bar that says "Ready".

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.

Creating or Editing a Policy

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

Follow the steps below to edit a policy:

- 1 On the toolbar, click **Policy Manager**
- or -
select **Tools** → **Policy Manager**.

The Policy Manager opens and loads, by default, the Standard policy.

- 2 To edit a policy that you previously created (i.e., a custom policy), select **File** → **Open** and select the policy.
- 3 To create a policy based on a prepackaged policy, 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 and choose **New Attack Group** from the shortcut menu. A highlighted entry named New Attack Group appears.
 - b Right-click the new group and choose **Rename**.
 - c Populate the group by dragging and dropping attack agents onto it.
- 7 You may also create a custom check. See [Creating a Custom Check](#) on page 227 for more information.
- 8 If you select the **Auto Update** check box, WebInspect determines if any updated or new attack agents downloaded from the HP 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 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.
- 9 Select **File** → **Save As**. Type a name for your custom policy in the **File name** box and then click **Save** to save the new policy in WebInspect’s *.policy format. You cannot save a policy using the name of a default WebInspect policy (Assault, Blank, Standard, etc.).

Creating a Custom Check

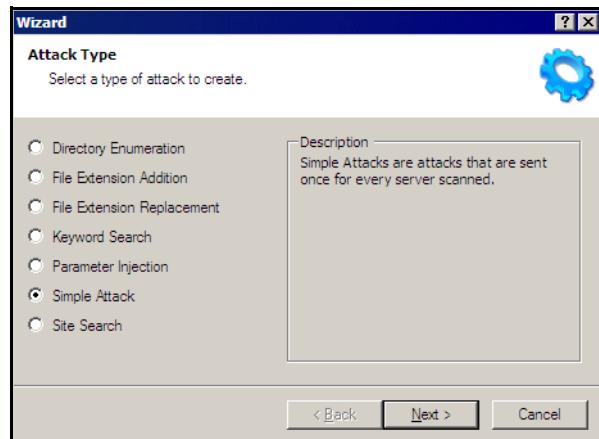
Although WebInspect rigorously inspects your entire Web site for real and potential security vulnerabilities, you may require a custom check to detect vulnerabilities that are unique to your application.

Follow the steps below to create a custom check:

- 1 On the WebInspect toolbar, click **Policy Manager**
- or -
click **Tools** → **Policy Manager**.

The Policy Manager opens and loads, by default, the Standard policy.

- 2 To edit a policy that you previously created, 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 Make sure the Standard view is selected, with attack groups listed in the left pane.
- 5 Right-click on **Custom Checks** and select **New Custom Check** from the shortcut menu.
- 6 When the Custom Check Wizard appears, select an attack type.



The attack types are listed below. See Steps 8-9 for entering attack and signature information.

- **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 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 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:\s+text/plain OR [HEADERS]Content-Type:\s+application/octet-stream)

- **File extension replacement**

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

For example, 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:\s+text/plain OR [HEADERS]Content-Type:\s+application/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\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 variations.

– 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\sbys\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: AAAAA...AAAAAA [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, WebInspect 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:

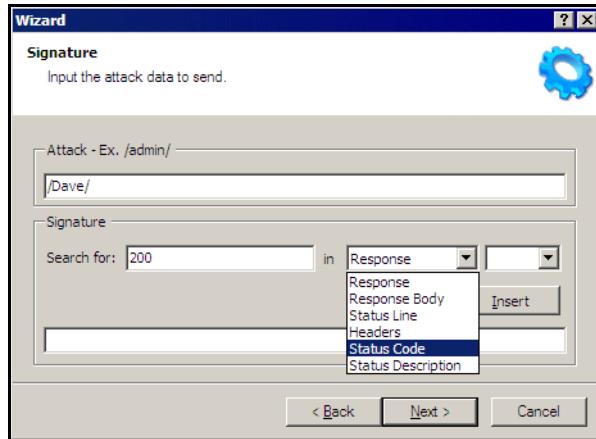
Attack Type: Site Search

Attack: confidential.txt

Signature: [STATUSCODE]2\d\d AND ([HEADERS]Content-Type:\text/plain
OR [HEADERS]Content-Type:\application/octet-stream)

7 Click **Next**.

8 In the **Attack** box, enter the data you want to use for the attack. In the following example of directory enumeration, the check will search for a directory named “Dave” by appending the attack string (/Dave/) to the target URL or IP address.



9 You must specify a signature, which is simply a regular expression (i.e., a special text string for describing a search pattern). When WebInspect searches the HTTP response and finds the text described by the signature, WebInspect flags the session as a vulnerability. You can use the **Search for** box 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** box:

a Enter the text you want to locate.

Enter only text; do not enter a regular expression.

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

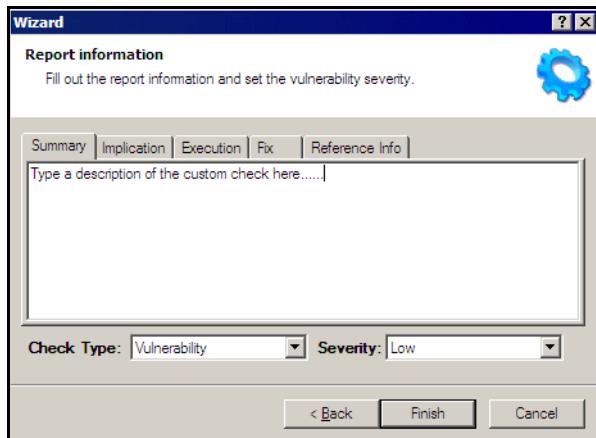
c Click the drop-down arrow to specify the section of the HTTP response that should be searched.

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

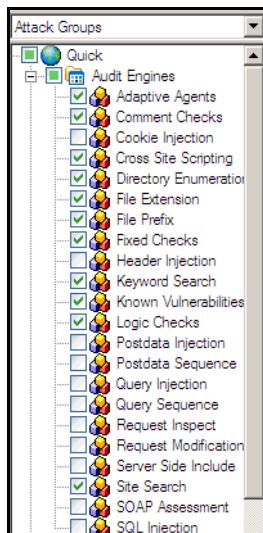
e Click **Insert**.

f (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**.



- 11 On the Report Information panel, click each tab and enter the text that will appear in the vulnerability 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 Click **⊕** to expand the Audit Engines folder.



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

Correlation of Attack Type to Audit Engine

This Attack Type...	Uses this Audit Engine...
Simple Attack	Fixed Checks
Parameter Injection	Post Data Injection
Site Search	Site Search

Correlation of Attack Type to Audit Engine (cont'd)

This Attack Type...	Uses this Audit Engine...
File Extension Replacement	File Extension
File Extension Addition	File Extension
Directory Enumeration	Directory Enumeration
Keyword Search	Keyword Search

18 Click **File** → **Save**.

19 Enter a name for the new policy and click **Save**.

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 226.

Disabling a Custom Check

Follow the steps below to disable a custom check:

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

Deleting a Custom Check

Follow the steps below to delete a custom check:

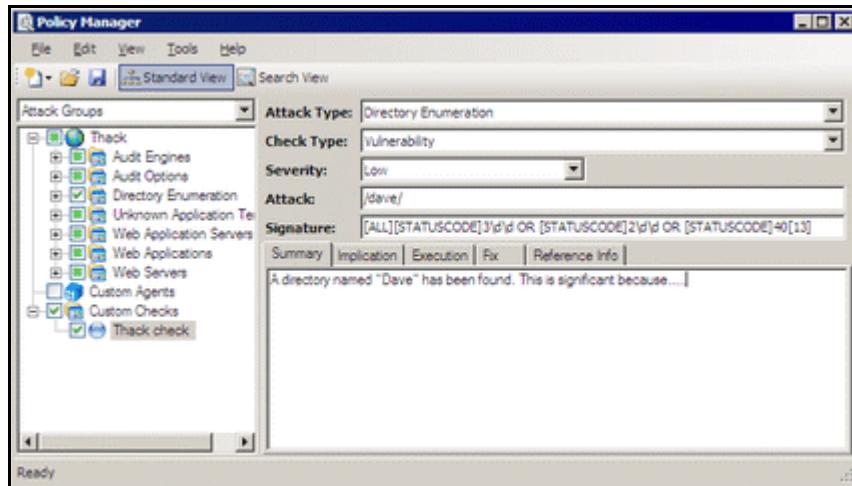
- 1** Right-click a custom check.
- 2** Select **Delete** from the short-cut menu.

Editing a Custom Check

Follow the steps below to edit a custom check:

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

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



- 4 Click the Save icon.

Searching for Attack 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.

Follow the steps below to search for attack agents:

- 1 On the toolbar, click **Policy Manager**
- or -
click **Tools** → **Policy Manager**.
- 2 If you do not have a policy selected, choose a policy from the *Open Policy* window and click **OK**.
- 3 Click **View** → **Search**.
- 4 From the **Criteria** list, select the property that you want to search.

The description of every attack agent contains “report fields” such as summary, implication, execution, fix, and reference information. The Search feature allows you to locate attack agents that contain the text you specify in a selected report field. In addition, you can search for a vulnerability ID, vulnerability name, engine type, or the date when last updated.

- 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**.

The Policy Manager lists in the **Checks** area all attack agents that match your search criteria. An active agent will have a check mark next to its name. Select (or clear) a check box to activate (or deactivate) an agent.

- 8 Click **Save** to save the revised policy.

Policy Manager Icons

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

Policy Manager Icons

Icon	Definition
	The policy.
	Attack Group Folder: Contains vulnerability assessments.
	Audit Methodology: A set of checks that compose an audit methodology. For example, Site Search is part of the Audit methodology.
	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 Inputs Editor

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's **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 Inputs 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 or Current Settings, the check inputs you create or modify become part of the settings file.

However, if you accessed the Audit Inputs Editor from the Policy Manager, you must import into WebInspect the saved file containing your check input modifications. To do so:

- 1 On the 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 and click **Open**.

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

Engine Inputs

Follow the steps below 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 WebInspect default 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
 - a To add an item to the list, click **Add**.
 - b To edit an item, select an item and click **Edit**.

- c To delete an item, select the item and click **Remove**.
 - d 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
 - a Unselect the **Use value from defaults** check box.
 - b Select an option from the drop-down list.
 - 6 Click **OK** (if you launched the Audit Inputs Editor from WebInspect's Default or Current Settings) or click the **File** menu and select **Save** or **Save As** (if you launched the Audit Inputs Editor from the Policy Manager).

Check Inputs

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

Follow the steps below to create or modify inputs for specific checks.

- 1 Click the **Check Inputs** tab.
- 2 Select a check (see list below).
- 3 Enter the requested input values.
- 4 Click **OK** (if you launched the Audit Inputs Editor from Default or Current Settings) or click **File** → **Save** (or **Save As**, if you launched the Audit Inputs Editor from the Policy Manager).

4719: IIS Mapping

Microsoft IIS extension handlers historically have been the source of many vulnerabilities. This check probes for each known IIS extension, and flags a vulnerability for each extension/handler that is found to be enabled. However, in certain cases, an extension handler may be legitimately enabled and used by the target Web site.

Required Input: One or more extensions that identify the handlers that are legitimately enabled and which should be excluded. Valid input is printer, idc, idq, ida, htr, htw, stm, shtm, and shtml.

[4721: Admin Section Must Require Authentication](#)

Any area of the Web site or Web application that contains sensitive information or access to privileged functionality (such as remote site administration) requires authentication before allowing access. This check attempts to access a sensitive directory that should require authentication. The default check input is /admin.

Required Input: The directory (relative to the root) containing administrative or sensitive data.

[4722: Logins Sent Over Unencrypted Connection](#)

Any area of a Web application that possibly contains sensitive information or access to privileged functionality (such as remote site administration functionality) should utilize SSL or another form of encryption to prevent login information from being sniffed or otherwise intercepted or stolen.

Required Input: Login forms. The name of file containing login form.

[4723: Logins Sent Over Query](#)

Information in query strings is directly visible to the end user via the browser interface, which can cause security issues. Recommendations include performing server-side input validation to ensure data received from the client matches expectations.

Required Input: Login forms. The name of file containing login form.

[4724: Password Field Masked](#)

Basic Web application security measures include “masking” all passwords entered by a user when logging on to a Web application. Normally, each character in a password entered by a user is instead represented with an asterisk. Recommendations include requiring all password fields in your Web application be masked to prevent other users from seeing this information.

Required Input: The name attribute of input controls containing a password.

[4726: Secure Section Only Accessible Via SSL](#)

Any area of the Web site or Web application that contains sensitive information or access to privileged functionality (such as remote site administration) requires that the pages under the secure section of the site are only accessible via SSL.

Required Input: The name of the secure directory, relative to the root. The default is /secure.

[4728: Persistent Cookies](#)

Persistent cookies are stored on the browser’s hard drive. This can cause security and privacy issues depending on the information stored in the cookie and how it is accessed. This check calculates how many seconds until a received cookie is set to expire. If the expiration date/time is less than the specified number of seconds (default: 600), the check considers the cookie’s life span to be excessive, increasing the chances of session ID recovery and session hijacking.

Required Input: The lifetime allowed for cookies (in seconds).

4729: User supplied data without POST

An area of the Web application that possibly contains sensitive information or access to privileged functionality (such as remote site administration functionality) uses query strings to pass information between pages. Information in query strings is directly visible to the end user via the browser interface, which can cause security issues. The input value for this check is a space-separated list of regular expressions that are used to identify sensitive URL parameter names when used in GET queries. Generally, information such as passwords, social security numbers, etc., should not be sent as parameters to GET queries, since the GET query (and thus the sensitive information) can persist in Web server and proxy logs and the Web browser's history. You will need to adjust the regular expressions accordingly to specify the parameter names your application typically uses to denote sensitive information.

Required Input: Sensitive parameter (a regular expression). An example is:

p | P]ass(word)? [u | U]ser_?([N | n]ame)? [s | S][s | S][n | N]

4731: Script Directory Check

A directory containing an object referenced in a post request or query string should not have a name that could easily be guessed by an attacker. The primary danger from an attacker discovering this directory would arise from the information he could gather from its contents, such as what language was used to code the Web application. This check is used to determine if a dynamic form action points to a file/URL that is in a directory whose name is included in the list.

Required Input: Names of directories containing scripts.

4732: Script File Extension Disclosure

Any area of the Web site or Web application that contains sensitive information or access to privileged functionality (such as remote site administration) requires the file extension of all scripts to be checked as it may lead to information disclosure related to the technology used by the application. The use of certain CGI-related file extensions can indicate certain types of technology in use, which results in a mild information disclosure. The default list of check input values is generally applicable, but some sites may legitimately use a certain technology (such as Perl) and this check may incorrectly elicit false-positive issues in flagging all Perl extensions (.pl). In such cases, you should remove the legitimate extensions from the list.

Required Input: File extensions of scripts used in the Web application (such as cgi, pl, and py).

5151: Arbitrary Remote File Include

This check attempts to discover if the Web application can fetch and incorporate data from arbitrary URLs supplied by an attacker.

This is the most complex check to configure, because its extreme flexibility enables it to work in many environments and topologies. Basically, the check injects URL values into application parameters, attempting to force the application to make an HTTP request to the supplied URL. This activity looks for “remote file inclusion” vulnerabilities caused by the application attempting to remotely retrieve the specified file/URL and include the response into the application’s processing. In certain extreme circumstances found in PHP environments, the application will remotely retrieve the file and execute any PHP script contained therein, making the activity capable of arbitrary code/script execution.

Check 5151 can operate in two modes: static and server (controlled by the “Audit Mode” parameter).

Static Mode

You specify the target external URL as the **Static Mode Target URL**, and a corresponding regular expression signature as the **Static Mode Signature**. If you want to use external targets, then you should use static mode. By default, the check uses static mode and the test URL of “<http://15.216.12.12/serverinclude.html?>” which is a special page hosted on an HP Web server located on the public Internet at IP address 15.216.12.12. The signature contains a specific value that is returned by the indicated test URL. If you do not want to use the HP Web server (particularly if the target server cannot access the Internet), then you should adjust the test URL (and corresponding signature) to a URL hosted by a server. When configuring static mode:

- Specify a full, absolute URL (i.e., it should begin with “<http://>”).
- For best results, use non-SSL URLs (although SSL URLs are allowed).
- Include a question mark (?) at the end of your URL to ensure the URL is not affected if the application appends additional data to the end of the URL.

Server Mode

In this mode, WebInspect runs its own Web server and attempts to get the target/scanned server to connect to the WebInspect scanning system. The added benefit of Server mode is that it can detect “blind” remote file inclusion vulnerabilities, resulting in potentially fewer false negatives. To use Server mode, the check conceptually needs three pieces of information:

Server Mode Target IP -- The IP address the server/target should use to access the host (particularly if the scanning system’s network IP is different than what the server would need to access, due to a firewall or a multi-homed scanning system). The default value is empty/blank, meaning that it uses the same IP address ultimately used or determined by the Server Mode Server IP.

- Server Mode Server Port -- The port number to run the listening Web server on. Using a specific port may be necessary due to network/access restrictions. The default value is 8181. If you leave this value blank, then the Remote File Include engine will dynamically choose a port between 25000 and 25100.
- Server Mode Server IP -- The local IP address of the scanning system to bind the Web server on, if the system is multi-homed and/or you do not want to bind the Web server listening on the first local IP address. The default value is “0.0.0.0”, which instructs WebInspect to use the first available IP address on the system.

Although the default values fit most configurations, certain circumstances require specific modification.

- If your system has multiple IP addresses (due to multiple network adapters), then you may need to specify the explicit IP address to bind to (i.e., the one that is most appropriate for receiving requests from the system you are scanning). You can determine the list of your system’s IP addresses by running “`biconvex`” from a Windows command prompt.
- If you are running multiple scans from the same scanning system using server mode, then you should leave the **Server Mode Server Port** value blank, causing WebInspect to dynamically pick the port. This is because two scans cannot run two separate Web servers listening on the same port. One specific port can only be used by one scan at a time.
- If your system is behind a firewall and you are using port-forwarding to receive the incoming HTTP requests, or you are on a network that uses NAT, then the IP address used by the server to access your system will be different from the IP address actually assigned to your system. In this case, you will need to specify the IP address the target server should use for the **Server Mode Target IP**.

Required Inputs: Static mode target URL, Audit mode (static or server), Server mode server IP, Server mode Server port, Server mode target IP, static mode signature (a regular expression)

5546: Privacy Policy Not Present

This check is associated with WebInspect's compliance policies. Many legislative initiatives require organizations to place a publicly accessible document within their Web application that defines their information privacy policy. If WebInspect does not find the specified file, it creates a vulnerability in the Best Practices category.

Required Inputs: The relative directory and file name of the privacy policy.

10167: Password in Query or Cookie Data

Transmitting password information in a query string or cookie value makes it easy for an attacker to see and tamper with login values. Recommendations include ensuring that login information is sent with a POST request over an encrypted connection and that sensitive account information is stored on the server.

Required Inputs:

- Password Field Names - List of Query or Cookie parameter names containing a password.
- Possible Username List - List of Query or Cookie parameter names containing a username.

10183: Allowed Top-Level Domain

Certain organizations (especially branches of the U.S. federal government) must use a restricted set of DNS top-level domain names (TLDs), such as .gov, .mil, or .fed. This check ensures that all allowed hosts encountered during the scan use one of the specified TLDs. Most public corporations arbitrarily use any TLD they desire (.com, .net, .org, etc.); those corporations should either disable this check (preferable) or change the default values to include .com, .net, and .org (and/or any other appropriate TLDs).

Required Inputs: All allowed top-level domains.

10274: Proxy CONNECT Access

Some proxy servers accept the CONNECT method to make an HTTP connection to another server. Usually, this method should be restricted to internal use only. If it is not restricted, your server can be used by an attacker on the Internet to disguise himself as your own server. Thus, any attack will appear to come from your server. This type of vulnerability is usually caused by not properly configuring the proxy server. Attackers can masquerade as your proxy server when conducting other attacks. Attackers may be able to access internal machines through the CONNECT proxy. This attack can also be used to enumerate your local network.

This check attempts to treat the target server as a proxy server for SSL requests. The check issues a CONNECT request to the target server, which essentially asks the server to make a connection to another external site. You can control which external site is used via the input values for this check. By default, the value "https://www.google.com/" is used, causing the server to make an external request to the host www.google.com on port 443. You may wish to modify this value to point to a more appropriate internal host. If so:

Use a server that has SSL enabled on the standard SSL port 443, if possible. Some proxies refuse connections to ports other than 443 due to explicit configuration.

Use the https:// URL format.

If you need to specify a port other than 443, use the normal URL format to specify a port after the host name (e.g., https://example.com:8443/).

Only the host name and port number are used; the remainder of the URL is ignored.

10275: Proxy GET Access

This check is virtually identical to check 10274, except it issues a proxy-qualified GET request to the target server instead of a CONNECT request. There are many servers that are willing to take a proxy-qualified GET request and treat it as a normal GET request (ignoring the proxy-specific aspects of the request), so it is necessary for the check to evaluate the response content to ensure the response is truly from the external server and not a normal response from the target. That is why check 10275 has two check inputs: one for specifying the external target host, and one for specifying a regular expression to match against the response content. By default the check attempts to access “http://www.google.com/” and looks for the phrase “Google Search” in the response. You will need to adjust the check input values if you need to use a different external host or an internal host. You can change the external target simply by adjusting the target check input value, and then specifying a unique value from the target page as the check input regex value.

- The URL target must begin with http:// or https://. For best results, use http://.
- If you need to specify a specific port other than 80/443, use the normal URL format to specify a port after the host name (such as http://example.com:8080).
- Unlike check 10274, the target URL you specify for 10275 is used in its entirety; if you specify a specific page/URL, then that specific page/URL will be requested.
- Try to select a unique value/phrase from the target URL to use as the response regex value, one that is not likely to appear elsewhere on the target scanned site; using the value in the <title> tags usually is sufficient (you can also include the “<title>” tags in the regex value itself).
- Remember to properly escape any regex-specific metacharacters (periods, parentheses, etc.).
- The check does not follow redirects (HTTP 302 responses), so you will need to specify an explicit final URL destination.

Required Inputs: Proxy GET target and Proxy GET target response (regular expression).

10280: Price-Related Form Fields

Forms containing price-related field names could harbor price manipulation vulnerabilities that would allow the attacker to change the price of the product.

Required Inputs: Names of price-related fields.

10287: Local File Include

Several types of attacks involve malformed filename requests that result in reading local files from the Web server. The Local File Include engine generates requests that contain variously encoded file names, and then evaluates the responses to determine if the contents of those files were recovered.

Mode

The Mode parameter relates to the platform assumptions made by the engine. The default mode value, **Auto**, causes the engine to look for both “c:\windows\win.ini” (Windows) and “/etc/passwd” (Unix) files and to use both Windows and Unix parent directory references accordingly. If the engine gets a visual response that explicitly indicates the underlying platform (Windows vs. Unix), it will automatically switch to using only the values for appropriate target platform for the remainder of the auditing for that application parameter value. If you already know what the underlying platform is before you scan (i.e., Windows vs. Unix), you can change the mode to **Windows** or **Unix**, which can save scan time since it reduces the number of values that need to be sent. At this time the engine does not support platforms that do not use a Windows (“\”) or Unix (“/”) path separator.

User-Specified File

If you want to use a specific target file, specify it here. There are occasions when the default file name values (“c:\windows\win.ini” and “/etc/passwd”) may not work in your environment. For example, your Web application can be hosted on a Windows drive other than ‘C:’, or your Web application could be operating out of a Unix chroot environment. In both cases, parent directory references will not be able to locate the specified target files even if a vulnerability does exist. For this situation you should either use an existing file that is in the root directory of the same drive/chroot of the Web application, or explicitly create a text file in the root directory of the drive/chroot used by your Web application and place a unique value inside the text file. Then you inform the LFI engine to look for your specific file by setting the **UserOnly** mode option, and specifying the absolute path to your target file in the **User Specified File** check input. You will also need to specify a corresponding **User Specified File Regex** check input value; the regex value should uniquely identify/match the contents of your specified file while not matching any content typically found on the scanned Web site. You can also select the **UserAndAuto** mode, which would let you specify a file and still use the default “c:\windows\win.ini” and “/etc/passwd” values.

User-Specified File Regex

If you use a specific target file, then you need to specify a regular expression that matches the contents of the target file.

Audit Disposition

The Audit Disposition parameter default value **Adaptive** treats Web application parameters in one of two ways: parameters with existing values that resemble file names receive significant (aggressive) scrutiny, while all other parameters receive basic scrutiny. The premise is that if the parameter has a value that resembles a filename, then there is a higher likelihood that the value is used in a file system operation; because of that higher likelihood, it makes sense for the engine to try more variations (particularly minor variations) to ensure that is not the case. However, trying additional minor variations can extend scan time, because it results in more attacks to be sent. That is why the **Adaptive** disposition tries to determine when it seems appropriate to spend the extra effort in auditing a particular parameter. However, if you desire the utmost level of scrutiny for all parameters, change the Audit Disposition value to **Aggressive**.

Required Inputs: Mode and Audit disposition.

10551: Possible Username or Password Disclosure

Exposing login information on publicly accessible sections of a Web Application could allow an attacker to access sensitive applications and information on a site, or to perform functions according to the privilege level of the login information. Gaining information critical to the success of escalated attacks would also be a likely impact of exploitation. Recommendations include purging the information from publicly accessible content, if possible, or otherwise ensuring proper access controls are in place.

Required Inputs:

- Password field names - Names of client-side script variables containing a password.
- Possible Username List - Names of client-side script variables containing a username.

10963: Cross-Site Request Forgery

Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a Web application in which he/she is currently authenticated. With a little help of social engineering (like sending a link via E-mail/chat), an attacker may force the users of a Web application to execute actions of the attacker's choosing. A successful CSRF exploit can compromise end user data and operation in case of normal user. If the targeted end user is the administrator account, this can compromise the entire Web application.

Criteria for identifying Cross-Site Request Forgery (CSRF)

- This check is only run against POST requests.
- The page must be either a login page or a page in restricted session (i.e., an authenticated session). Note: To avoid testing every POST request made during authenticated sessions, the check is run against a URL one time. This means that forms with multiple parameters will be tested one time only and not multiple times like a cross-site scripting or parameter injection check.
- The page is not a re-authentication page. This is to avoid cases where a user is asked to either change a password or provide a password when already in an authenticated session. A re-authentication page is not CSRF vulnerable.
- The page does not contain CAPTCHA. A CAPTCHA page is not vulnerable to CSRF.
- The page is not an error page or an invalid page from the server.

Check inputs are used as heuristics to help the CSRF agent refine detected results. There are a number of criteria used for CSRF detection that help to avoid false positives.

Required Inputs

- Password field names - This field is used to help identify login pages. The matches are string matches.
- Possible Username List - This field is used to help identify login pages. The matches here are string matches.

Optional Inputs

- CSRF Request Black List - This field is used to identify pages that are NOT to be flagged as vulnerable to CSRF. Matching values are identified for the name values in POST parameters.

- CSRF Response Black List - This field is used to identify error pages or invalid pages. The default value here is a combination of two regular expressions and also a string value (CAPTCHA). Matching values are identified on the response body.
- CSRF Response White List - This field is used to elevate the risk associated with this vulnerability for specific pages. By default, CSRF findings are a Medium severity. A match for values in this field will result in the finding being rated as a High severity. Matching values are identified in the response body.

10965: User Data in Query or Cookie

Transmitting password information in a query string or cookie value makes it easy for an attacker to see and tamper with login values. Recommendations include ensuring that login information is sent with a POST request over an encrypted connection and that sensitive account information is stored on the server.

Required Inputs:

- Password Field Names - List of Query or Cookie parameter names containing a password.
- Possible Username List - List of Query or Cookie parameter names containing a username.

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 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 login form, the user will proceed to the application’s beginning page.

Some sites (such as WebInspect’s example banking application zero.webappsecurity.com) contain many different forms for completing a variety of transactions. If WebInspect is 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 WebInspect encounters an input control whose name attribute is not matched in the file you create, it will submit a default value (12345).

► If you are using a proxy server, the WebForm Editor will not use the default settings from WebInspect. You must first configure Internet Explorer to use the desired proxy.

There are two ways to create a list of form values:

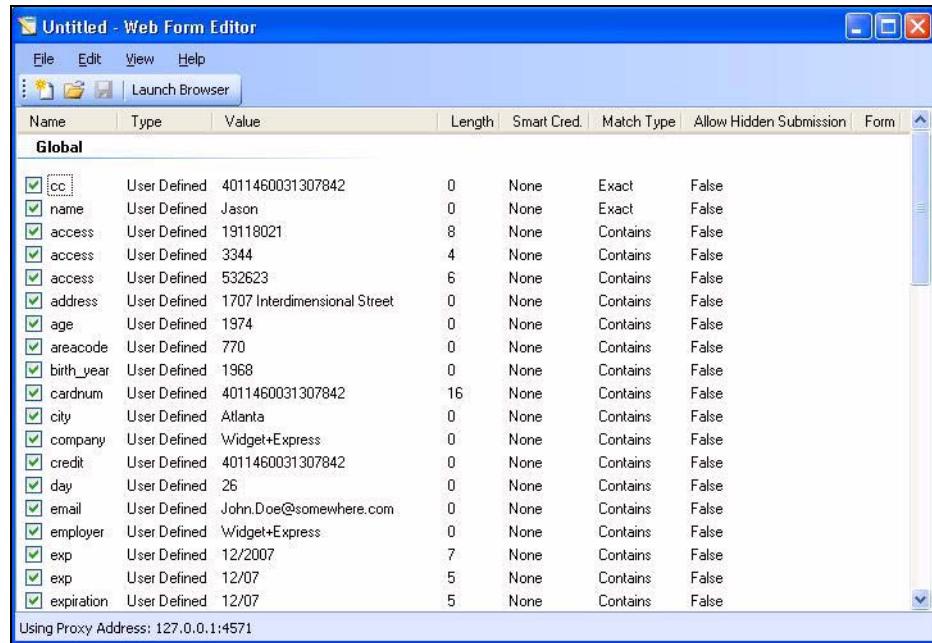
- Create the list manually.
- Record the values as you navigate through the application.

Manually Creating a Web Form List

Use the following procedure to create a Web Form list manually.

- 1 Click **Tools → WebForm Editor**.

The *WebForm Editor* window appears.



The WebForm Editor loads a prepackaged default file.

- a To load a different file, select **File → Open**.
- b To create a new file, select **File → New**.
- 2 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 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.

- 3 In the **Name** box, type (or modify) the name attribute of the input element.
- 4 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).

- 5 In the **Value** box, type the data that should be associated with the input element (for example, a password).
- 6 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.
- 7 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**.
- 8 Click **Add (or Modify)**.
- 9 If necessary, you can assign additional attributes by right-clicking an entry and using the shortcut (pop-up) menu.
- To submit the form value for any input control having the same name attribute, regardless of the URL at which it occurs, choose **Make Global**.
 - 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**.

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.”

- If you select **Mark As Interactive Input**, 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**).

It is not necessary to tag passwords with **Mark As Interactive Input**.

Recording 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 clicking **Edit → Settings**.

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 **File → New** (or click the New icon on the toolbar).
- 2 To add form values to an existing list, select **File → Open** (or click the Open icon on the toolbar) and choose a file using the standard file-selection dialog.
- 3 Click **Launch Browser**.

- Using the browser's **Address** bar, enter or select a URL and navigate to a page containing a form.

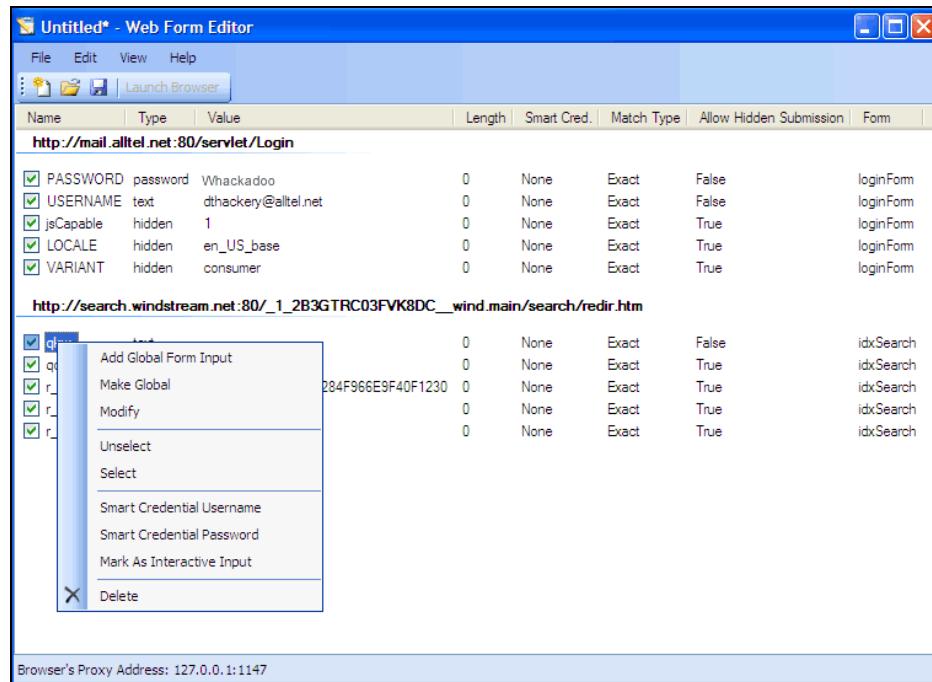
Note: Internet Explorer version 7 and the .NET Framework are hard-coded not to send requests for localhost through any proxies. Therefore, the Web Form Editor will not receive such traffic. This is a documented Microsoft defect. To access a site on "localhost" when using IE7, place a period or dot after "localhost" (for example, http://localhost.:8080/test.html).

- Complete the form and submit it (usually by clicking a button such as **Log In**, **Submit**, **Go**, etc.).
- Navigate to additional pages and submit forms until you have traversed all the links you wish to follow.
- The Web Form Editor displays a list of name and value attributes for all input controls found in all forms on the pages you visited.

For example, the first two entries in the following illustration 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.



- If necessary, you can modify items by right-clicking an entry and using the shortcut (pop-up) menu.
 - To submit the form value for any input control having the same name attribute, regardless of the URL at which it occurs, choose **Make Global**.
 - 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**.
- To force the scanner to pause 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 will pause the scan and display 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, WebInspect will 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).

- 9 Click **File → Save (or Save As)**.

Importing a Web Form File

You can import a file that was designed and created for earlier versions of WebInspect and convert it to a file that can be used by the current Web Form Editor.

- 1 Click **File → 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**.

Scanning with a Web Form File

When scanning a site, you specify which Web Form file you want to use by selecting **Auto-fill web forms during crawl** (step 3 of Perform a Web Site Scan) and then selecting a file.

You can also designate a specific file as the default by using the following procedure:

- 1 On the WebInspect menu bar, click **Edit → Default 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) 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 Web Form Editor's **File** menu and select **New**.
 - c Click **Launch Browser**.
 - d See [Recording Web Form Values](#) on page 248 for further instructions.
- 6 To edit Web form values for the selected file:
 - a Click **Edit Current Web Form Values**.
 - b See [Recording Web Form Values](#) on page 248 for further instructions.

Web Form Editor Settings

Follow the steps below to modify the Web Form Editor settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **General** or **Proxy** category and enter the settings described in the following sections.
- 3 Click **OK**.

General

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 IP address and port by selecting **Edit → Settings**.

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 by selecting an entry from the **Assumed 'charset' Encoding** list.

Proxy

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

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 Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

HTTPS Proxy Settings

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

Web Form Logic

When crawling a Web application and submitting Web form values, WebInspect 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

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 wildcard.	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 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 for the record allows for submission to any field length (wildcard field length match).

Rules for Matching Web Form Values (cont'd)

	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."

Web Brute

This tool will determine if your users are employing user names and passwords that an unauthorized intruder might be able to guess easily. For example, if one of your customers is accessing your Web site by using a username of “customer” and a password of “password,” you might want to warn that user about his susceptibility and suggest that he change his password and/or username.

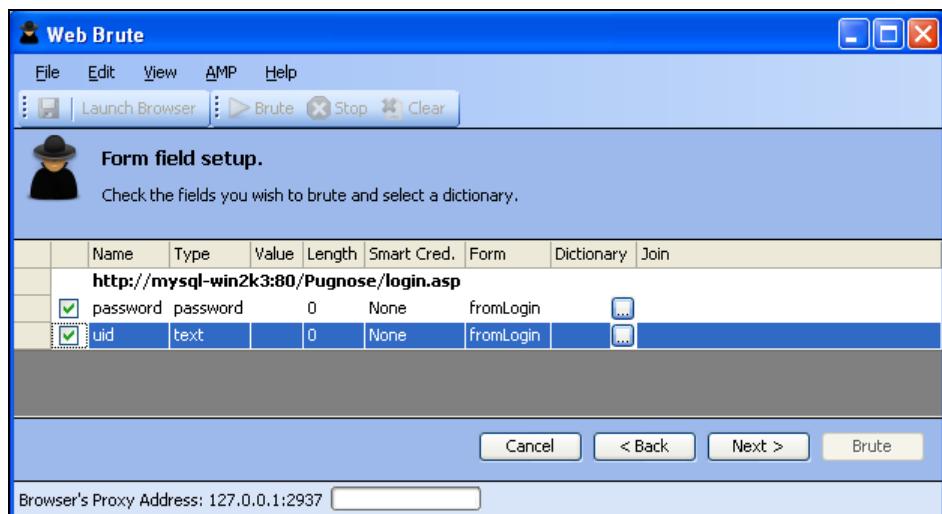
Web Brute will attempt a “brute force” attack of a login form or authentication page, using two prepared lists of user names and passwords.

- !!** This is an intrusive attack and can break into a secure area. Brute force attacks are intended for testing purposes only, and should not be used against unsuspecting Web sites.

Mounting a Brute Force Attack

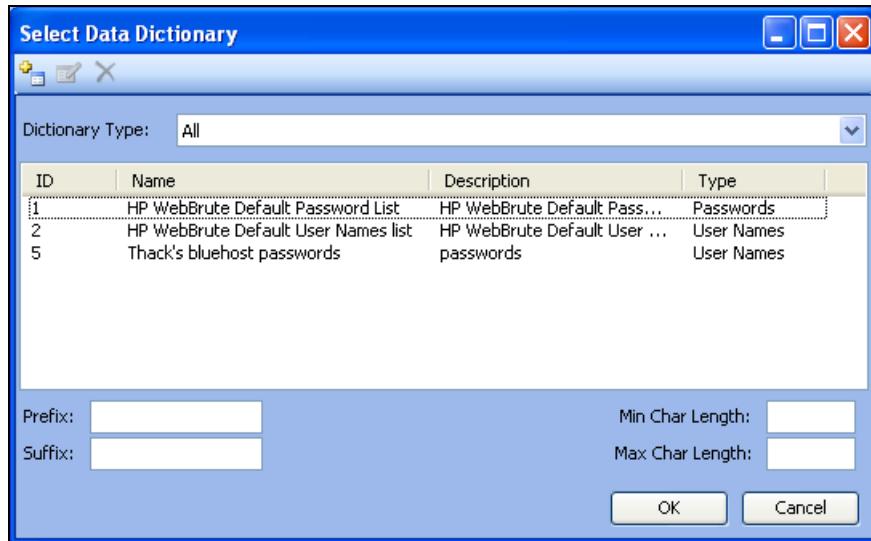
Follow the steps below to use a brute force authentication attack:

- 1 On the WebInspect menu bar, click **Tools** → **Web Brute**.
- 2 In the **Enter URL** box, type the URL of the site you want attack and click **Next**.
- 3 Select the authentication type used by the target site. See Authentication Types on page 177 for a description of the available authentication types.
- 4 If necessary, use the **Domain** box to specify the domain that should be used for authentication. Web Brute will prefix this string to each user name that it submits. Do not include a backslash.
- 5 Click **Next**.
- 6 If you selected **Web Form** in Step 3, a Web browser opens. If necessary, navigate to the login page.
- 7 On Web Brute’s **Form Field Setup** panel, select (check) the fields you want to brute force. If you already know the value that should be entered for a field, remove the check mark, double-click the cell in the **Value** column for that field, and enter the value.



- 8 For fields you have selected (checked), click  in the **Dictionary** column to select a list of names or passwords to be submitted.

The *Select Data Dictionary* window appears, listing all currently defined dictionaries. You can limit the display of dictionary names by selecting an entry in the **Dictionary Type** list.



These dictionaries are in a database that is not directly accessible. To create your own dictionary or merge a list into an existing dictionary, see [Creating and Importing Lists](#) on page 257.

- 9 Select a list.
- 10 (Optional) Enter the following:
 - **Prefix**—A string that will be added to the beginning of each entry in the list.
 - **Suffix**—A string that will be added to the end of each entry in the list.
 - **Min Char Length**—The minimum number of characters allowed for each entry; entries that are shorter will not be submitted.
 - **Max Char Length**—The maximum number of characters allowed for each entry; entries that are longer will not be submitted.

- 11 Click **OK**.
- 12 Repeat steps 7-11 for each authentication field to be submitted.
- 13 If you want to “join” two or more lists, click the **Join** column associated with each list.

If a list of user names is joined with a list of passwords, then Web Brute will submit user names with passwords in the order in which they appear in the lists. That is, the first name in the user name list will be submitted with the first password in the password list, the second name will be submitted with the second password, etc.

If the two lists are not joined, then Web Brute submits each user name with all passwords. This feature is used most often for Web form authentication where the user must re-enter the password. In this case, Web Brute would use two lists, but the password list would be specified for both the “password” and “confirm password” fields. You would then join these fields, forcing the same password to be submitted for each field.

- 14 To modify the parameters that Web Brute uses during an authentication attack, select **Edit → Settings**. See [Web Brute Settings](#) on page 258 for more information.

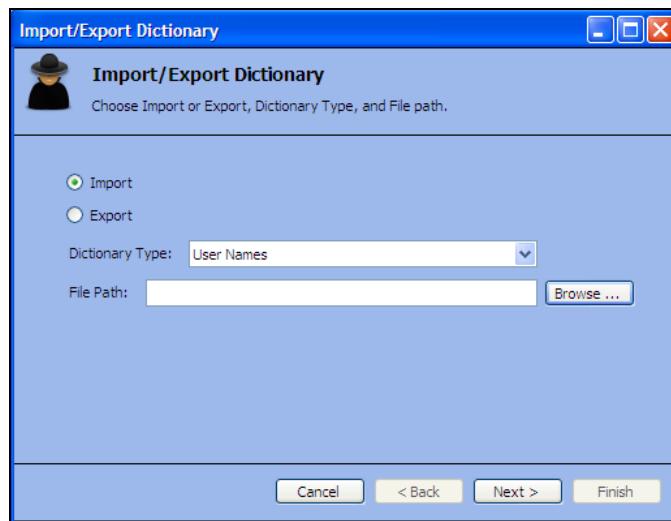
- 15 Click **Next**.
- 16 To see a list of failed name/password attempts (in addition to successful attempts), select **Show Failed**.
- 17 Click **Brute**.

Web Brute attacks the site and displays the results. If you double-click a result (either successful or failed), Web Brute opens the HTTP Editor, allowing you to inspect both the HTTP request and response.

Creating and Importing Lists

To use your own list of passwords or user names, you must first create a list and then import it into Web Brute as a “dictionary,” using the following procedure:

- 1 Create a text file where each entry is delimited by a carriage return and line feed.
- 2 Click **File → Import/Export Dictionary**.
- 3 On the *Import / Export Dictionary* window, select **Import**.



- 4 From the **Dictionary Type** list, select either **User Names**, **Passwords**, or **E-mails**.
- 5 Click **Browse** and select the file containing the list you want to import.
- 6 Click **Next**.
- 7 On the *Import Dictionary* window, specify a name for the dictionary and enter a description.
- 8 Click **Next**.
- 9 Click **Finish**.

Exporting Dictionaries

Use the following procedure to create a text file from a Web Brute dictionary:

- 1 Click **File → Import/Export Dictionary**.
- 2 On the *Import / Export Dictionary* window, select **Export**.

- 3 In the **File Path** box, enter the path and name of the text file in which the dictionary contents will be saved, or click **Browse** and use the *Save As* window to specify the name and path.
- 4 Click **Next**.
- 5 On the *Export Dictionary* window, select a dictionary type from the list.
- 6 Select a dictionary.
- 7 Click **Next**.
- 8 Click **Finish**.
- 9 Click **Done**.

Web Brute Settings

Follow the steps below to modify the Web Brute settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **Options**, **Authentication**, or **Proxy** tab and enter the settings described in the following sections.
- 3 Click **OK**.

Options

Timeout in seconds

Enter the number of seconds that Web Brute will wait for a response. If a response is not received during this period, Web Brute will resend the request, up to the number of times specified in the Retry Count setting.

Retry Count

Enter the number of times that Web Brute will resend a request that has not been acknowledged.

Apply State

If you select this option, Web Brute will attempt to maintain state during the procedure.

Apply Proxy

If you select this option, Web Brute will use the settings on the Proxy tab to connect to the target site (if the Direct Connection option is not selected).

Logging

Select the types of messages that should be logged.

Max Concurrent Threads

Enter or select the number of requests that Web Brute may send before requiring a response to the first request.

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 Brute should use.

Authentication

If required, select an authentication method and provide credentials. The methods are:

- **None**—Select this option if the site does not require authentication.
- **Automatic Authentication**—This allows Web Brute to determine the correct authentication type.
- **HTTP Basic Authentication**—This is a widely used, industry-standard method for collecting user name and password information. Normally, a 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.
- **NTLM Authentication**—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.

After selecting an authentication method, enter a user name and password. To avoid typographic errors, re-enter the password in the **Confirm Password** box.

Proxy

Use these settings to access the Web Brute through a proxy server.

[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 Internet Explorer proxy settings](#)

Import your proxy server information from Internet Explorer.

[Use Firefox proxy settings](#)

Import your proxy server information from Firefox.

[Configure a proxy using a PAC file](#)

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

Specify Alternative Proxy for HTTPS

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

Web Discovery

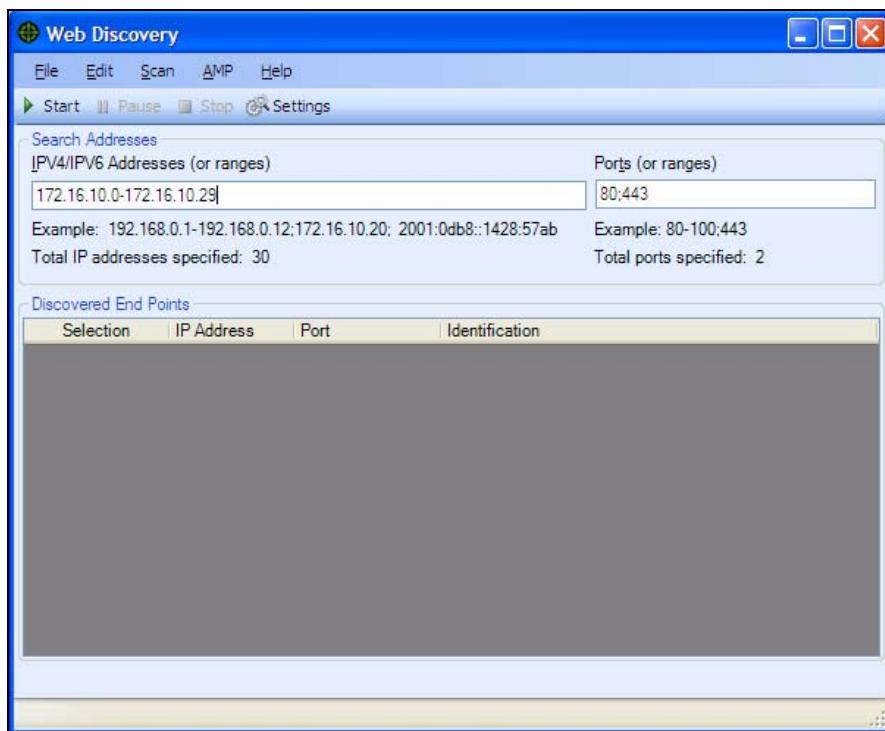
Use Web Discovery to find all open hosts in your enterprise environment.

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.



Discovering Sites

To discover sites using Web Discovery:

- 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. See [Internet Protocol Version 6](#) on page 150.

- 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 [Web Discovery Settings](#) on page 262 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 *Settings Properties* window and view the raw request and response.
- To save the list of discovered servers:
- 1 Click **File → Export**.
 - 2 Use the standard file-selection window to name and save the file.

Web Discovery Settings

Follow the steps below to modify the Web Discovery settings:

- 1 Click **Edit → Settings**.
- 2 Enter the settings described in the following sections.
- 3 Click **OK**.

Select Protocols

Choose the packet type you want to send by selecting or clearing the check box next to the protocol name.

Logging

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.

Enter the file location in the **Log To** box, or click the browse button  and use the standard file-selection window to specify the file in which the log entries should be recorded.

Connectivity

Set the following timeouts (in milliseconds):

- **Connection**—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**—When sending a message to the remote IP endpoint, 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**—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.

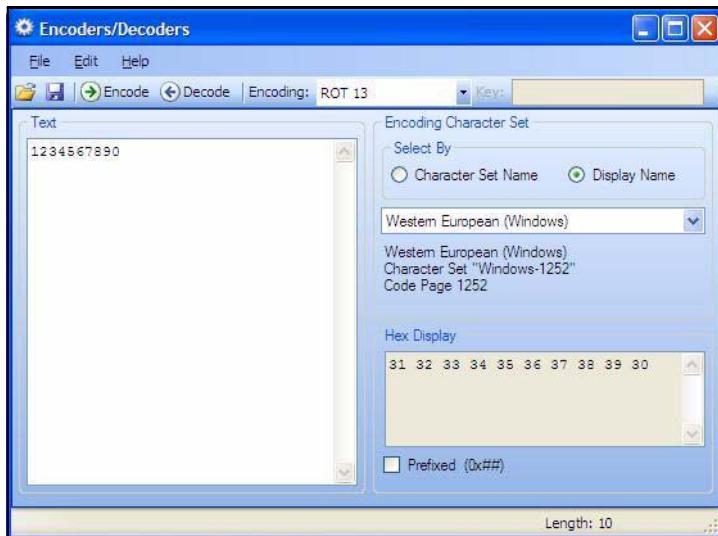
Adjust the number of open sockets using the **Sockets** box. A higher number of open sockets results in a faster scan. However, a setting that exceeds a server's threshold may result in false positives



If you are using Windows XP with Service Pack 2 (SP2), your **Open Sockets** setting is set to 10. If you increase this setting, you will increase the likelihood that the scanner will fail to detect servers. This occurs because Microsoft has limited the number of open sockets to 10.

Encoders/Decoders

This tool allows you to encode and decode values using Base64, 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 a String

Follow the steps below to encode a string:

- 1 Type (or paste) a string into the **Text** area, or load the contents of a file by selecting **File** → **Open**.
- 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 265.
- 4 If necessary, type a key in the **Key** box. 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).

Decoding a String

Follow the steps below 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**.
- 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.

- 4 If necessary, type a key in the **Key** box.
- 5 Click **Decode**.

You can also use 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 three 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** 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 below.

- 3DES is 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 is an encryption algorithm that can be used as a replacement for the DES algorithm.
- DES (Data Encryption Standard) is 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 is hexadecimal.
- MD5 produces a 128-bit “fingerprint” or “message digest” of whatever data you enter.
- RC2 is 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 is 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 is a simple Caesar cipher used for obscuring text by replacing each letter with the letter thirteen places down the alphabet.

- SHA1 is 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 uses 256-bit encryption.
- SHA384 uses 384-bit encryption.
- SHA512 uses 512-bit encryption.
- ToLower changes upper-case letters to lower-case.
- ToUpper changes lower-case letters to upper-case.
- TwoFish is 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 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.

Prefixed

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.

Regular Expression Editor

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.

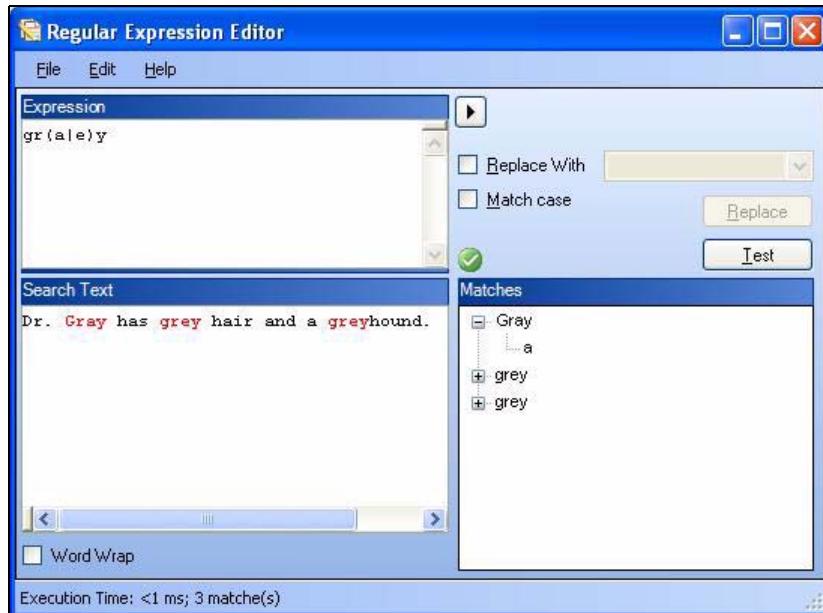
Testing a Regular Expression

Use the Regular Expression Editor to verify regular expressions.

Follow the steps below to use the Regular Expression Editor:

- 1 Click Tools → Regex Editor.

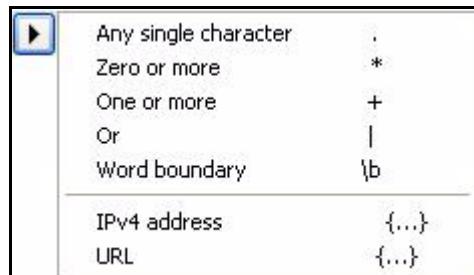
The *Regular Expression Editor* window opens.



- 2 In the **Expression** box, type or paste a regular expression that you believe 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 special 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** box, 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. To do so:

- 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 the 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 If you want 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 will be highlighted in red.

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

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.

Characters Used in Regular Expressions

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: <code>/content/([^\ec].* e[^n].* c[^a].* .{3,})[/./]*</code> 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.”

Characters Used in Regular Expressions (cont'd)

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 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

Hewlett-Packard 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 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
```

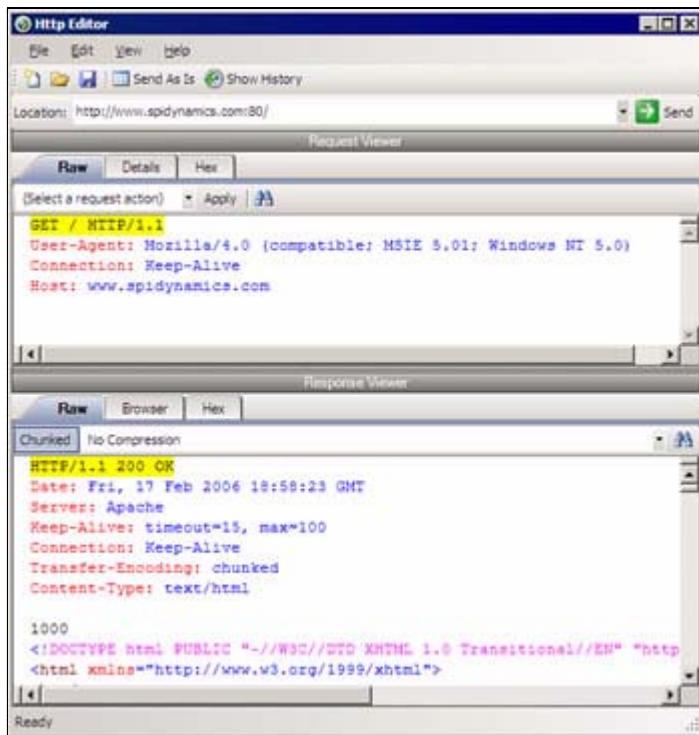
To detect a response containing a specific string (such as “Please Authenticate”) in the Reason-Phrase portion of the status line, use the following regular expression:

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

HTTP Editor

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 that requires a working knowledge of HTML, HTTP, and request methods.

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



Request Viewer

The Request Viewer pane contains the HTTP request message, which you can view in three 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 (Note: This tab appears only if the request contains XML-formatted data).

Response Viewer

The Response Viewer pane contains the HTTP response message, which you can also view in three 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 (Note: This tab appears only if the response contains XML-formatted data).

HTTP Editor Menus

File Menu

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

- **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 most standard HTTP proxy servers cannot process non-compliant HTTP requests.

- **Exit**—Closes the HTTP Editor.

Edit Menu

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

- **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 window 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 commands:

- **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 Web sites offers significant cost savings over the use of legacy character sets. Unicode enables a single software product or a single Web site 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** list on the Request 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 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* window, 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].

Editing and Sending Requests

Follow the steps below to edit and send a request.

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

To change certain features of the request, select an item from the **Action** list and click **Apply**.

- 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**).

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

Searching for Text

Follow the steps below 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**.

HTTP Editor Settings

Follow the steps below to modify the HTTP Editor settings:

- 1 Click **Edit** → **Settings**.
- 2 Select either the **Options**, **Authentication**, or **Proxy** category and enter the settings described in the following sections.
- 3 Click **OK**.

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 may 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 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 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 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 from 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.

Enable Active Content

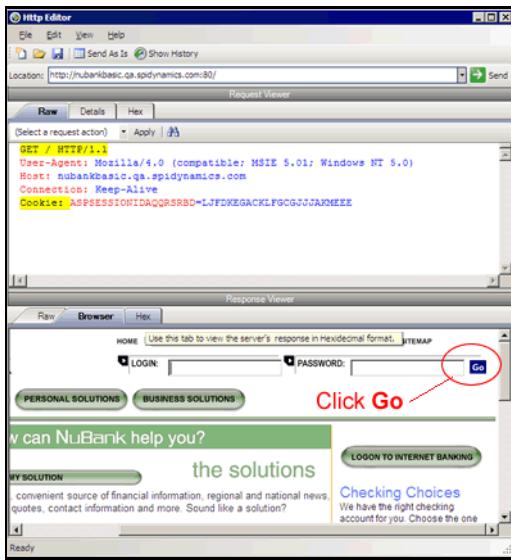
Select this option to allow execution of JavaScript and other dynamic content in all browser windows.

Navigation

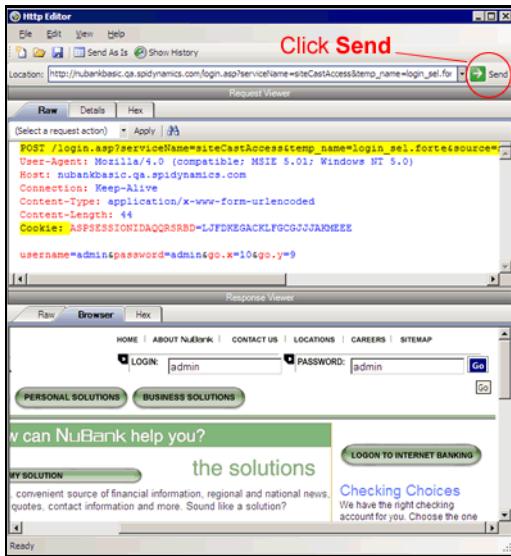
In the **Navigation** group, select either **None**, **Interactive**, or **Browser Mode**.

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.

For example, using the logon page at nubankbasic.qa.spidynamics.com (shown below), you could enter a user name (“admin”) and password (“admin”), and then click **Go**.



The HTTP Editor formats the request (which uses the POST method to the login1.asp 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**.



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.

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 HTTP Editor should use.

Authentication

If authentication is required, select a type from the **Authentication** list. See Authentication Types on page 177 for a description of the available authentication types.

After selecting an authentication method, enter a user name and password. To avoid typographic errors, re-enter the password in the **Confirm Password** box.

Proxy

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

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's Web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering 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. See Authentication Types on page 177 for a description of the available authentication types.
- 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.

HTTPS Proxy Settings

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

Web Proxy

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 WebInspect, a browser, or any other tool that submits HTTP requests and receives responses from a server. It is a tool for debugging and penetration scanning; you can see every request and server response while browsing a site.

You can also create a Startup macro or a Login macro that you can use with WebInspect.

Before using Web Proxy with your browser, you must configure your browser's proxy settings. If using Internet Explorer:

- 1 Click **Tools → Internet Options**.
- 2 Click the **Connections** tab.
- 3 Click **LAN Settings**.
- 4 On the *LAN Settings* window, select **Use a Proxy Server for your LAN** and enter the address and port of the proxy server you want to use. By default, Web Proxy uses your local host settings (127.0.0.1:8080).



Note: Internet Explorer version 7 and the .NET Framework are hard-coded not to send requests for localhost through any proxies. Therefore, Web Proxy will not receive such traffic. This is a documented Microsoft defect. To access a site on “localhost” when using IE7, place a period or dot after “localhost” (for example, `http://localhost.:8080/test.html`).

You should also configure Microsoft Internet Explorer to use HTTP 1.1 through proxy connections. On Internet Explorer:

- 1 Click **Tools → Internet Options**.
- 2 Click the **Advanced** tab.
- 3 In the “HTTP1.1 settings” section, select **Use HTTP 1.1 through proxy connections**.

Using Web Proxy

Follow the steps below to use Web Proxy with a browser:

- 1 Click **Tools → Web Proxy**.

The *Web Proxy* window opens.

- 2 Click or select **Proxy → Start**.

“Listening on <server:port number>” displays in the Web Proxy status bar.

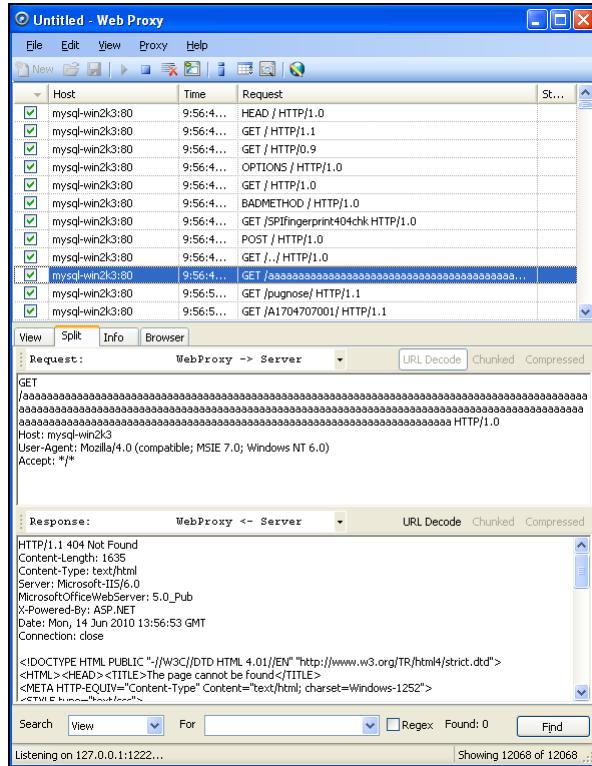
- 3 Click **Launch Browser** .

This starts a Web browser and configures it to communicate through Web Proxy.

- 4 Manually navigate the site for which you want to view requests/responses.

Internet Explorer version 7 and the .NET Framework are hard-coded not to send requests for localhost through any proxies. Therefore, Web Proxy will not receive such traffic. This is a documented Microsoft defect. To access a site on “localhost” when using IE7, place a period or dot after “localhost” (for example, `http://localhost.:8080/test.html`).

- 5 If Web Proxy receives a request for a certificate from a Web Server, it displays a dialog 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 click **Proxy → Stop**).
- 7 Each session (a request and matching response) you recorded is listed in the top pane. To view the actual HTTP message, select an entry. The message appears in the bottom frame. By default, the **View** tab is selected.



- 8 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 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%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.

- 9 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).
- 10 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  (or 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 WebInspect scan. All **File** menu commands apply to “check-marked” requests.

Click the top of any column to sort the requests by that selection. For example, to sort the requests by the time they were made, click the top border of the **Time** column.

- You must stop Web Proxy when you want to change Web Proxy settings.

Creating a Web Macro

You can use either the Web Macro Recorder or Web Proxy to create a Start macro or a Login macro.

A Start macro is used most often to focus on a particular subsection of an application. It specifies URLs that an HP 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.

Follow the steps below to create a 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 **File** → **Create Web Macro**.
- 3 (Optional) On the *Create Web Macro* window, 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.

Background: 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 scan. 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 condition, the scanner searches every response message for this phrase. Whenever it detects the phrase, the scanner attempts to log in again by sending an HTTP request containing the username 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 background 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.

- 4 Enter a name in the **Save macro as** box.
- 5 Click **OK**.

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.

Web Proxy Tabs

Tab	Description
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: <ul style="list-style-type: none">• 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). You can cut, paste, and copy the raw request, and right-click to see a shortcut menu of encoding options. However, you cannot save an edited request from the Web Proxy tool. Use the HTTP Editor to save an edited request.
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.

Web Proxy Tabs (cont'd)

Tab	Description
Browser	Click the Browser tab to view the response as formatted in a browser.

Web Proxy Settings

To access this feature, click **Edit** → **Settings**.

- You cannot change settings while Web Proxy is running. Click **Proxy** → **Stop**, change settings, and then restart Web Proxy.

Task 1: Configure General Settings

- 1 Select the **General** tab.
- 2 In the **Proxy Listener Configuration** group, 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.

Both Web Proxy and your Web browser must use the same IP address and port. If using Internet Explorer, click **Tools** → **Internet Options**; click the **Connections** tab and click **LAN Settings**; on the *LAN Settings* window, select **Use a Proxy Server for your LAN** and enter the address and port of the proxy server you want to use.

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.

- 3 Use the **Do Not Record** option to create a regular expression filter that prevents 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.
- 4 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 when you click the **Allow** button.

- 5 In the **Logging** group, 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.

- 6 Most Web pages contain information that tells the browser what language encoding to use. This is accomplished by using a META tag with an HTTP-EQUIV attribute in the HEAD section of the HTML document, as in the following example:

```
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
```

For pages that do not announce their character set, choose an option from the **Assumed 'charset' Encoding** list to select the language (and implied character set) that Web Proxy should use.

Task 2: Configure Proxy Servers Settings

- 1 Click the **Proxy Servers** tab.

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).

- 2 In the **Proxy Address** box, type the IP address of the server through which you want to route Web Proxy requests.
- 3 Specify the port number in the **Proxy Port** box.
- 4 Select the type of proxy (standard, SOCKS4, or SOCKS5) from the **Proxy Type** list.
- 5 If this proxy server requires authentication, select an authentication type and enter your authentication credentials in the **Username** and **Password** boxes. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 6 Click **Add** to add the server and display its IP address in the **Available Proxy Servers** list.

You can also import a file containing a list of proxy servers by clicking **Import**. The file containing proxy information must be formatted as follows:

- Each line contains one record followed by a line feed and carriage return.
- Each field in the record is separated by a semicolon.
- The fields appear in the following order: address;port;proxytype;user name;password.
- The user name 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

127.153.0.3;80;socks4;;

128.121.6.9;443;socks5;myname;mypassword

- 7 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.
 - a Click **Add** in the **Bypass Proxy List** group.

The *Bypass Proxy* window appears.

 - b 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.*

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 resolves to the IP address that you specify, Web Proxy will still send the request to a proxy server.

- c Click **OK**.

Task 3: Configure Search-and-Replace Settings

- 1 Click the **Search and Replace** tab.

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

- 2 Click **Add** to create a default entry in the table.
- 3 Click the **Search Field** column of the entry.
- 4 Click the drop-down arrow and select the message area you want to search.
- 5 In the **Search For** column, type the data (or a regular expression representing the data) you want to find.
- 6 In the **Replace With** column, type the data you want to substitute for the found data.
- 7 Repeat this procedure to create additional search rules.

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



The request/response rules are applied sequentially, in the order in which they appear. For example, if a rule changes HTTPS to SSL, and a subsequent rule then changes SSL to SECURE, the result will be that HTTPS is changed to SECURE.

Task 4: Configure Flag Settings

- 1 Click the **Flag** tab.

- This feature allows you to find and highlight keywords in requests or responses.
- 2 Click **Add** to create a default entry in the table.
 - 3 Click the **Search Field** column of the entry.
 - 4 Click the drop-down arrow and select the message area you want to search.
 - 5 In the **Search** column, type the data (or a regular expression representing the data) you want to find.
 - 6 Click the **Flag** column of the entry.
 - 7 Click the drop-down arrow and select a color with which to highlight the data, if found.

Task 5: Configure Evasion Settings

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.



This feature is intended for use as a penetration testing tool. Do not use it or enable it when conducting vulnerability scans with WebInspect.

Use the following procedure to enable evasions:

- 1 Select the **Evasions** tab.
- 2 Select **Enable Evasions**.

Choose one or more evasion techniques, as described below.

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/ HTTP/1.1
```

Web Proxy sends the following message to the server:

```
HEAD http://www.microsoft.com/ 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<CRLF><CRLF>
```

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<CRLF><CRLF>
```

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 /YPVIFAH[D 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 device 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
```

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 **Allow**. 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 **Proxy** tab on the *Web Proxy Settings* window, you can force Web Proxy to pause after each request, after each response, or after locating specific text in either the request or response.



Follow the steps below to turn on interactive mode:

- 1 Click **Proxy** → **Stop**.
- 2 Click **Proxy** → **Interactive**
-or-
- 3 Click **Proxy** → **Start**.

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. Clicking the icon or selecting the command will toggle the Interactive mode on or off.

Smart Update

Use Smart Update to download the latest adaptive agents and programs, as well as vulnerability and policy information. Smart Update also ensures that you are using the latest version of WebInspect, and prompts you if a newer version of the product is available for download. New vulnerability checks downloaded via Smart Update are not added automatically to any custom policies you may have created.



Caution: For AMP Installations, if Smart Update changes or replaces certain AMP-related files used by WebInspect, the sensor service may stop and the sensor will display a status of “off line.” You must launch the WebInspect application and restart the service. To do so, choose Configure from the AMP menu, and then click the **Start** button on the **Sensor Service** tab of the *AMP Configuration* window.

- 1 From the toolbar, click **Smart Update**
- or -
select **Smart Update** from the **Tools** menu
- or -
select **Start Smart Update** from the WebInspect Start Page.
- 2 If updates are available, the *Smart Update* window displays up to three separate collapsible panes for downloading the following:
 - New and updated checks
 - WebInspect software
 - Smart Update softwareSelect the check box associated with one or more of the download options.

- 3 To install the updates, click **Download**.

If you download checks without also downloading available new versions of WebInspect, HP will continue to offer updates to your installed knowledgebase for only 10 days. Beyond that period, updates will not be available to you until you download the new WebInspect software.

Checking for Updates Automatically

You can force WebInspect to check for new vulnerabilities or program components every time you open the application. This is the simplest method of ensuring that your SecureBase vulnerabilities database remains accurate and includes the latest information. To enable this option, select **Application Settings** from the **Edit** menu and choose **Smart Update**.

Cookie Cruncher

The Cookie Cruncher analyzes cookies to determine the relative ease with which an attacker could predict or determine the value of a session ID generated by a server and delivered to a client via a cookie.

Background

The Web's Hypertext Transfer Protocol (HTTP) is stateless, meaning that each communication is discrete and unrelated to those that precede or follow. Because there is no continuity inherent in the protocol, application designers introduced the concept of "session." A session is defined as all activity by a user with a unique IP address on a Web site during a specified period of time. When a user logs into an application, a session is created on the server to maintain the state for other requests originating from the same user.

Each session has a unique identifier (session ID). This text string is transmitted between the client and the server, and may be stored in cookies, URLs, or hidden fields of Web pages. One problem with session IDs, however, is that many Web sites generate them using algorithms based on easily predictable variables, such as time or IP address. This predictability makes the Web sites vulnerable to session hijacking.

Session hijacking involves an attacker using session IDs to seize control of a legitimate user's session while that session is still in progress. The attacker can then gain complete access to the user's data, and can perform all operations that are normally available to the legitimate owner of the session.

Using the Cookie Cruncher

Follow the steps below to use the Cookie Cruncher:

- 1 In the **URL** box, enter the URL of the site you want to test.

If you are using the Cookie Cruncher to examine a site you have scanned with WebInspect, follow these steps:

- a In the WebInspect navigation pane, click the cookies icon  Cookies. All HTTP responses containing a "Set-Cookie:" header are listed in the information pane.

- b Double-click one of the listed responses.

- c Click **Request**.

- d Copy the request and paste it into the Cookie Cruncher's **Request** area.

- 2 In the **Sample** box, enter the number of requests the Cookie Cruncher should send to the server (expecting a cookie to be returned). A higher number of samples increases processing time, but produces more reliable result; a minimum of 100 is suggested.

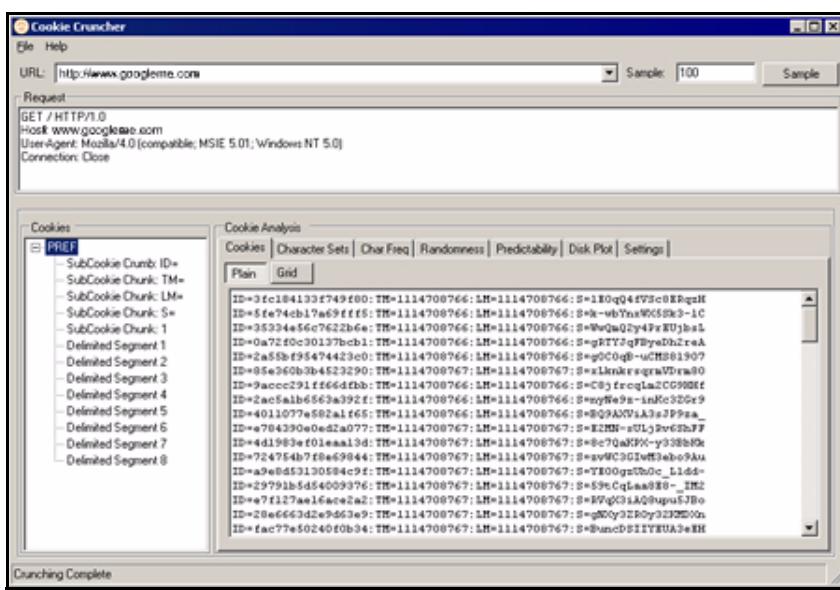
- 3 Click **Sample**.

As cookies are collected, the Cookie Cruncher organizes them into a tree hierarchy displayed in the vertical pane on the left side of the window.

- 4 Click a cookie in the tree hierarchy to analyze it. If subcookies are found, the Cookie Cruncher modifies the tree hierarchy; click the plus sign  to expand the level. Repeat as necessary.

- 5 To view the analysis, select a cookie or subcookie and click the various tabs.

- 6 To save the sampled cookies for future analysis, click **File → Save**.
- Cookie Cruncher cannot open and display a saved cookie file (.sck) if it contains fewer than four cookies.



Subcookies

Subcookies are either portions of cookie values that are common to many cookies, or interpreted values.

When the same string of characters appears in multiple cookies, you can choose that as a subcookie. The recurring expression will be eliminated from the cookies that contain it, and those cookies will be re-analyzed. The portion that is removed (the recurring expression) is called a “subcookie crumb.”

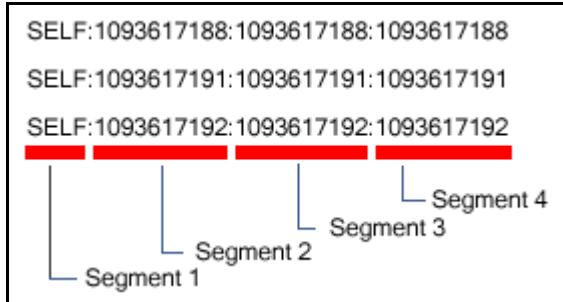
In the following sample, “086-” would be detected as a recurring expression:

```
086-1123
086-1127
087-6281
086-1132
088-0518
087-6282
```

Analysis of those cookies containing the recurring expression (1123, 1127, 1132) would reveal the (most likely) incrementing cookie values that were interleaved with values from some other source.

If the detected character set of a sample consists of just 10 characters (Q-Z), these characters could possibly represent the digits 0-9. Choosing the re-encode option would run the cookies through an appropriate decoder algorithm (base-10, base-16, base-64, etc.) and re-analyze the cookies.

The “Delimited Segment” option(s) allow you to select the delimited portions of cookies. For example, the following subcookies contain four delimited segments.



To analyze the second segment of all subcookies, you would click the **Select Subcookie** list and select **Delimited Segment 2**.

For more information, see the white paper [Automated Cookie Analysis](#).

Cookie Cruncher Tabs

Use the Cookie Cruncher tabs to analyze the sampled cookies. The tabs are:

- Cookies
- Character Sets
- Char Freq
- Randomness
- Predictability
- Disk Plot

Cookies Tab

This tab lists all cookies received from the server. You can view them either in plain or grid format by clicking the appropriate button.

Character Sets Tab

This tab displays the character set used to format the cookie:

- A = alphabetic character (letters A-Z)
- N = numeric character (numbers 0-9)
- H = hexadecimal character (0-F)
- T = Text A-Z, a-z
- I = Illegal (anything else)
- D = delimiter

Char Freq Tab

This tab displays a graph showing the number of times each ASCII character appeared in the total sample of cookies. A pale blue dot indicates an ASCII character whose number of appearances equals the number of cookies. A highlighted character indicates that it may be a delimiter (which is usually a character such as a comma, colon, or semicolon, but could also be something unusual such as “Z”).

Randomness Tab

This tab attempts to differentiate between random and non-random portions of cookies, based on the sample obtained.

Use the Grid view to illustrate the analysis of each column. The color key is:

- Red = No randomness (or very little)
- Orange = Somewhat random
- White = Random

The top row of the grid indicates the numeric position of each character.

The second row displays, for each character position, a number representing the relative randomness of the character. This is actually the average number of bits that change per column from one cookie to the next.

Use the Graph view to illustrate the randomness level in a graphic format. The dashed green line represents the optimum (best practice) level of randomness. The red line represents the randomness of the cookies in the sample. In a well designed cookie, the red line should follow the green line. When the graph view is selected, you can save the graph (in BMP, GIF, PNG, or JPG format) using the **Save Graph** command in the **File** menu.

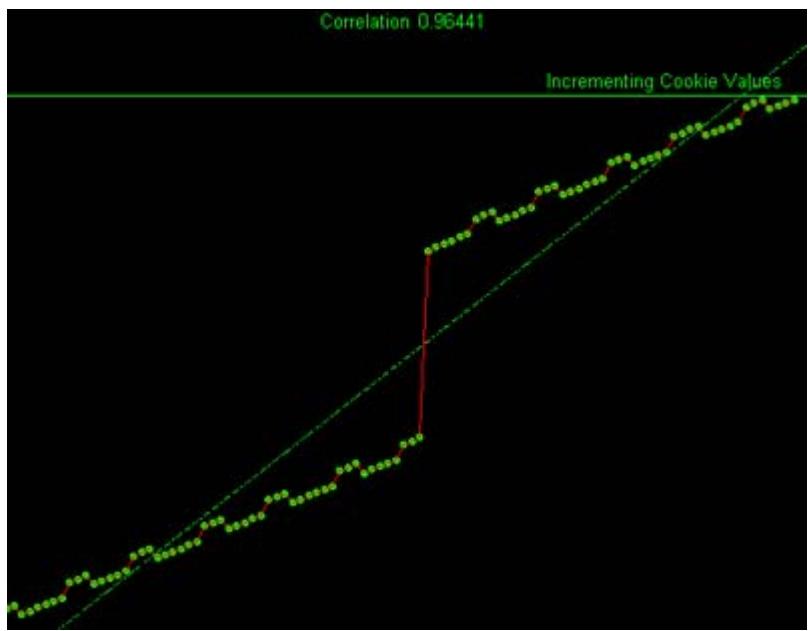
Predictability Tab

The Cookie Cruncher analysis produces a correlation value ranging from 0 to 1 and displays it at the top of the graph. A low value indicates that cookie generation is more random; a higher value indicates greater predictability.

The value of each cookie is plotted (on the Y axis) against the time the cookie was received (on the X axis). A scattered distribution indicates randomness, whereas a pattern approaching a line indicates predictability.

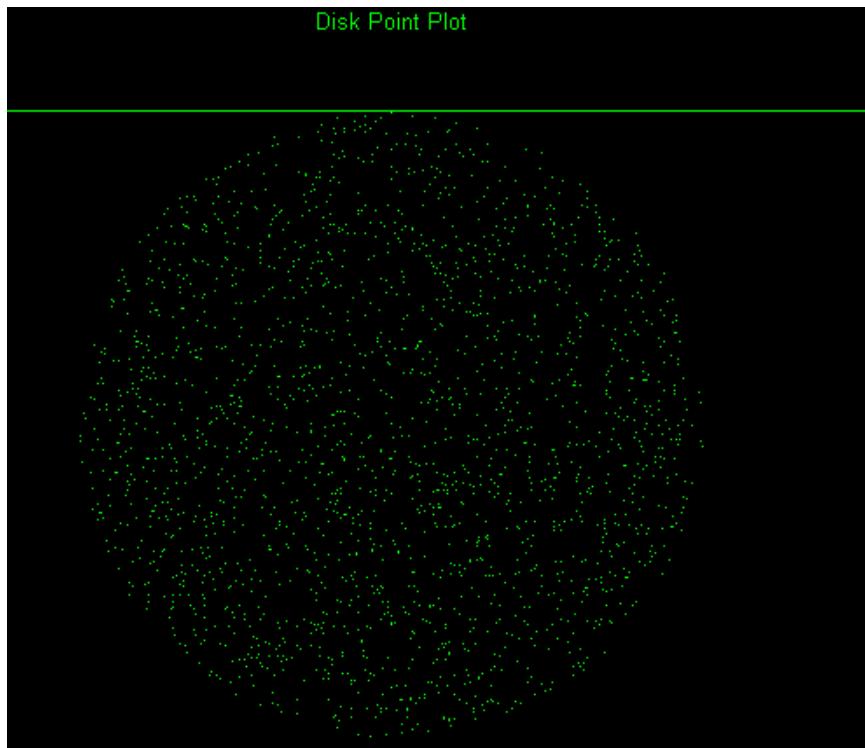
If the correlation is .9 or greater, the graph displays the header “Incrementing Cookie Values” or “Decrementing Cookie Values” and draws a “best fit” line.

Only decimal or hexadecimal values can be plotted.



Disk Plot Tab

This graph plots a cookie's value against the sine and cosine functions. When random data is plotted, the points are evenly distributed around the plotting area. Only decimal or hexadecimal values can be plotted.



Cookie Cruncher Settings

Follow the steps below to modify the Cookie Cruncher settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **General**, **Authentication**, or **Proxy** category and enter the settings described in the following sections.
- 3 Click **OK**.

General

Thread Count

Specify the maximum number of threads that can be created. The Cookie Cruncher can be configured to send up to 75 concurrent HTTP requests before waiting for an HTTP response to the first request. The default setting is 10. Increasing the thread count will increase the speed of the process, but might also exhaust your system resources as well as those of the server you are scanning. While most servers can handle a large number of requests, servers in development environments sometimes have licensing limitations that allow only five or fewer users to be connected at a single time. In such cases, you should reduce the thread count to be less than 5.

Socket Timeout

Specify the maximum number of open sockets permitted. A higher number of open sockets results in a faster process. However, a setting that exceeds a server's threshold may result in false positives.

If the Cookie Cruncher runs on Windows XP with Service Pack 2 (SP2), the number of open sockets should be set to 10.

Custom Delimiters

The Cookie Cruncher interprets certain characters (such as `/.-!;:=`) as delimiters. In some cases, you may want to substitute your own list. For example, a cookie having a value of "ABC123456-C:Program" contains two default delimiters — a dash (-) and a colon (:) — and would therefore be split into three parts. However, if you specify only the dash as a delimiter, the cookie would be split into just two parts.

The user-specified list, if present, will cause an extra subcookie type to appear in the tree, in addition to the regularly parsed subcookie types. The subcookie item may not appear when the number of cookies having the delimiter(s) is less than 10 percent of the total cookie sample.

To create a list of custom delimiters, select the **Parse with Custom Delimiters** check box and then enter one or more delimiters in the **Characters** box.

Authentication

Authentication Method

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

Authentication	Description
Automatic	Allow the Cookie Cruncher to determine the correct authentication type. Note that 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. 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.
NT LAN Manager (NTLM)	NTLM is an authentication process used by all members of the Windows NT family of products. It 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.

Authentication Credentials

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

Proxy

Use these settings to access the Cookie Cruncher through a proxy server.

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 Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

[Use Firefox proxy settings](#)

Import your proxy server information from Firefox.

[Configure a proxy using a PAC file](#)

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

[Explicitly configure proxy](#)

Configure a proxy by entering 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

[HTTPS Proxy Settings](#)

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

Web Fuzzer

“Fuzzing” is an automated software testing technique that generates and submits random or sequential data to various areas of an application in an attempt to uncover security vulnerabilities. For example, when searching for buffer overflows, a tester can simply generate data of various sizes and send it to one of the application entry points to observe how the application handles it.

The Web Fuzzer lets you run several automated tests for common classes of Web application security vulnerabilities such as SQL injection, format strings, cross-site scripting, path traversal, odd characters, and buffer overflows, as well as protocol implementation problems.

Using the Web Fuzzer

Follow the steps below to use the Web Fuzzer:

- 1 Click **Edit → Server**.
- 2 Enter the fully qualified domain name or IP address of a Web site, along with other server configuration information, and click **OK**.
- 3 Click **Edit → Settings**.
- 4 Configure the settings and click **OK**. For more information, see [Web Fuzzer Settings](#) on page 305.
- 5 To create a session, click **Session** and select either **Create** or **Raw Create**.
 - a If you select **Create**, Web Fuzzer displays a tabbed property sheet that identifies each section of an HTTP request and allows you to replace an HTTP element with generated data or with text that you enter. This structured approach is recommended for novice users. For detailed information, see [Using the Session Editor](#) on page 302.
 - b If you select **Raw Create**, Web Fuzzer displays a standard GET request formatted as regular text. You can edit the request. You can also place the cursor anywhere in the request, right-click to invoke a shortcut menu, and then insert a generator that will fuzz the selected HTTP element. If you highlight any portion of the request, the highlighted portion will be replaced by the generator.

Fuzzer Generators

Generator	Function
Number	Inserts a whole number, within the range you specify, in each request; you specify the starting and ending number, the increment, and the number of times to loop through the series.
ASCII	Inserts one ASCII character, within the range you specify, in each request; you specify the starting and ending character, and the number of times to loop through the series.
Character	Generates the character you specify and inserts multiple numbers of the character into each request; you specify the minimum and maximum number of characters, and an increment.

Fuzzer Generators (cont'd)

Generator	Function
Decimal Number	Inserts a fractional number, within the range you specify, in each request; you specify the starting and ending number, the increment, and the number of times to loop through the series.
Guid	Inserts a random Globally Unique Identifier (a 128-bit number) in each request; you specify the number of requests.
WordList Reader	Inserts a string from a text file you specify; the number of requests is determined by the number of paragraphs in the file; all characters in the paragraph are inserted.
SQL Injection	Inserts a string from a text file you specify; the number of requests is determined by the number of paragraphs in the file; all characters in the paragraph are inserted. The default file (sqlinjections.txt) contains the following two entries: ' or 1=1 ' or like '%
Text	Inserts the text you specify in a single request.
Cross-Site Scripting	Inserts a string from a text file you specify; the number of requests is determined by the number of paragraphs in the file; all characters in the paragraph are inserted. The default file (xssinjections.txt) contains the following entry: <script>alert('test')</script>
Method	Inserts a method (GET, POST, PUT, etc.); you specify the protocol version (0.9, 1.0, 1.1, or all).

- 6 After creating the request, click **OK**.
- 7 You can use filters so that only those server responses meeting criteria you specify will be displayed.
- 8 On the *Web Fuzzer Request* window, click **Start**.
The **Sessions** area lists each session (request and response) generated by the tool.
- 9 To examine the results, click an entry in the **Sessions** list.
 - The HTTP request for the selected session appears in the **Request** area.
 - The server's response appears on both the **Browser View** and **Raw Response** tabs.
- 10 To edit the request that you constructed, select a session in the **Sessions** group, then click the **Session** menu and choose either **Edit** or **Raw Edit**.

Filters

A filter consists of a name, description, and rule. The rule is a regular expression that defines the text you want to locate in a particular section of the server's response. For example, if you want to display only those responses that contain the word "error" in the response body and where the response also specifies a status code between 500 and 599, then use the following rule:

```
[STATUSCODE]5\d\d AND [BODY]\serror\s
```

Use the following notation to specify a response section:

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

You access the **Filters** window by selecting **Filters** → **Edit**.

In addition to enabling a specific rule, you must also enable the use of rules in general by selecting **Filters** → **Enable**.

Creating a Filter

Follow the steps below to create a filter:

- 1 Click **Add**.
The tool creates a rule named Default Rule.
- 2 Modify the Name, Description, and Rule.
- 3 Click **Apply** to save the definition.

Using a Filter

Follow the steps below to use a filter in a session:

- 1 Select a filter from the **Filters** list.
- 2 Select the **Enable** check box.

Deleting a Filter

Follow the steps below to delete a filter:

- 1 Select a filter from the **Filters** list.
- 2 Click **Delete**.

Editing a Filter

Follow the steps below to edit a filter:

- 1 Select a filter from the **Filters** list.
- 2 Modify the Name, Description, or Rule.
- 3 Click **Apply** to save the modifications.

Using the Session Editor

Use this tabbed property sheet to change specific sections of an HTTP request. You can replace an HTTP element with text that you type or paste into a text box, or you can insert a generator that will create multiple requests containing generated data.

Follow the steps below to use the Session Editor:

- 1 Click a tab.
- 2 You can either:
 - Edit the data appearing in text boxes, or
 - Select the **Use Generator** check box and click **Generator** to insert a generator.
- 3 To change other areas, click a different tab.
- 4 After configuring the areas you want to change, click **OK**.
- 5 When you return to the *Web Fuzzer* window, click **Start**.

Creating a Query String

Follow the steps below to create a query string:

- 1 Click **Add**.

The text “name=value” appears in the list, representing the query string you are creating.
- 2 Click the **Name** tab.

You can edit the parameter named “name” or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
- 3 Click the **Separator** tab.

You can edit the character that separates the name from the value (usually an equals sign) or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
- 4 Click the **Value** tab.

You can edit the value in the equation or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
- 5 Click the **Format** tab.

You can edit the order in which the equation elements appear, or you can introduce characters between them.
- 6 In the **Name Value Separator** group, you can edit the character that separates parameters (usually an ampersand) or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
- 7 To add another parameter, click **Add** and repeat Steps 2-6.

Session Editor Tabs

Method Tab

The GET method is specified by default. You can replace it with any text, or you can insert the Method generator.

Path Tab

You can fuzz three elements related to the path: the name of the file, the file extension, and the character that designates a directory level (usually the forward slash /). You can replace these elements with any text, or you can insert generators.

Query Tab

Some HTTP requests include a query string, with each parameter formatted as parameter=value and separated by an ampersand (&). The resource is separated from the query by a delimiter character (usually a question mark, although other characters can be used depending on the application). For example:

http://www.website.com/category.cfm?model_ID=0&category_ID=12.

Version Tab

The version indicates to the server which HTTP version to use for interpreting the request. Valid versions are 0.9, 1.0 and 1.1. The version information is formatted as “HTTP/version,” which is a name-value pair separated by a forward slash (/). You can fuzz all three sections: Protocol, Separator, and Version. You can also fuzz the format by rearranging the order or introducing extraneous characters.

Headers Tab

Headers contain basic information issued by the client to help the server or application handle the request. Common headers are Host and User-Agent. Each header is defined by using the “name: value” syntax. This name-value structure also can be separated into four fuzzing opportunities.

Creating Headers

Follow the steps below to create headers:

- 1 Click **Add**.

The text “name:value” appears in the list, representing the header you are creating.

- 2 Click the **Name** tab. You can edit the parameter named “name” or you can substitute a generator for it (select the **Use Generator** check box and click **Generator**).
- 3 Click the **Separator** tab. You can edit the character that separates the name from the value (usually a colon) or you can substitute a generator for it.
- 4 Click the **Value** tab. You can edit the “value” text or you can substitute a generator for it.
- 5 Click the **Format** tab. You can edit the order in which the header elements appear, or you can introduce characters between them.

- 6 In the **Name Value Separator** group, you can edit the character that separates headers or you can substitute a generator for it.
- 7 To add another header, click **Add** and repeat Steps 2-6.

Cookies Tab

Cookies are special headers that contain parameters used by the application to manage users and states. The format of a cookie definition is:

Cookie: name=value;name=value

Each parameter is a name-value pair that can be independently fuzzed.

Creating Cookies

Follow the steps below to create cookies:

- 1 In the **Cookies** group, click **Add**.
“Cookie.” appears in the list, representing the cookie you are creating.
- 2 Click **Cookie:** (in the Cookies list) and then click **Add** (in the **Cookie** group).
The text “name=value” appears.
- 3 In the **Cookie** group, click the **Cookie Name** tab. You can edit the name or you can substitute a generator for it.
- 4 Click the **Separator** tab. You can edit the character that separates the name from the value (usually an equals sign) or you can substitute a generator for it.
- 5 Click the **Value** tab. You can edit the “value” text or you can substitute a generator for it.
- 6 Click the **Format** tab. You can edit the order in which the header elements appear, or you can introduce characters between them.
- 7 In the **Name Value Separator** group, you can edit the character that separates headers or you can substitute a generator for it.
- 8 To add another cookie, repeat steps 1-7.

Post Data Tab

While a query can be appended to the Request-URI, post data is added to the end of the request. The format is similar to the URI query and is mostly used with the POST method. When post data are used, the request must contain a Content-Length header that indicates the size of the post data. You can fuzz not only the post data, but also the Content-Length value to test how the server or application handles the differences.

When fuzzing the HTTP request message, you affect two main layers of the application environment: server protocol implementation and Web application.

Creating POST Data

Follow the steps below to create post data:

- 1 Click **Add**.
The text “name=value” appears in the list, representing the post data you are creating.
- 2 Click the **Name** tab. You can edit the parameter named “name” or you can substitute a generator for it.

- 3 Click the **Separator** tab. You can edit the character that separates the name from the value (usually a colon) or you can substitute a generator for it.
- 4 Click the **Value** tab. You can edit the “value” text or you can substitute a generator for it.
- 5 Click the **Format** tab. You can edit the order in which the header elements appear, or you can introduce characters between them.
- 6 In the **Name Value Separator** group, you can edit the character that separates headers or you can substitute a generator for it.
- 7 To add another post data element, click **Add** and repeat Steps 2-6.

Web Fuzzer Settings

Follow the steps below to modify the Web Fuzzer settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **General** or **Proxy** category and enter the settings described in the following sections.
- 3 Click **OK**.

General

Enable Filters

Select this option to enable filter support.

Auto scroll view

Select this option to enable automatic scrolling in the **Sessions List** view. This will force the view to scroll down to the latest session automatically.

Show ToolTips

Select this option to enable the display of tool tips when you hover your mouse pointer over certain controls.

Sockets

Enter the maximum number of sockets and the sockets send timeout (in seconds).

Protocol Compliance

Select **Enforce Content-Length** to automatically adjust the Content-Length value in the request when needed. If this feature is enabled, you cannot fuzz the content-length header.

Select **Enforce Host header** to include the Host header in all requests. If this feature is enabled, you cannot fuzz the host header.

Proxy

Use these settings to access the Web Fuzzer through a proxy server.

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 Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

HTTPS Proxy Settings

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

SQL Injector

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 as database types and also supports multiple language systems including Japanese.

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 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.

Using the SQL Injector

Follow the steps below 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. See [SQL Injector Settings](#) on page 309 for additional information.
- 2 Select **File** → **New**
- or -
click the New Request icon.
- 3 In the **Location** box, 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`

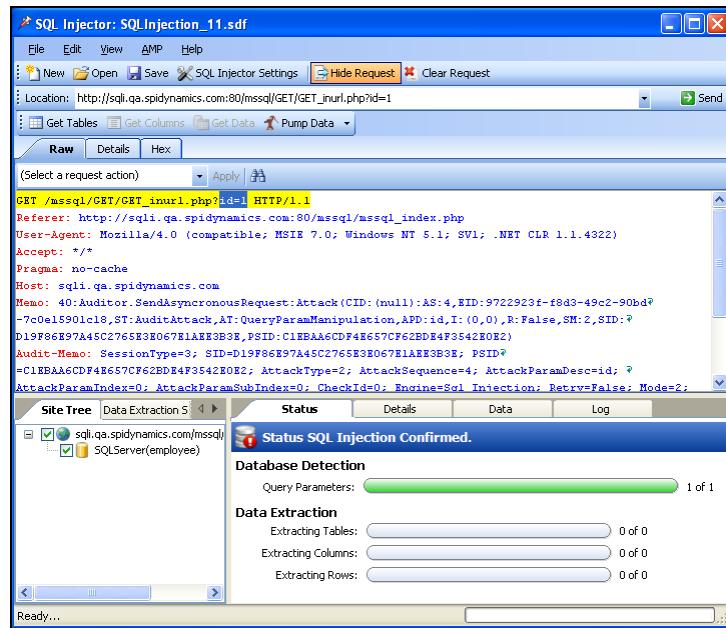
Because the SQL Injector defaults to the GET method, you must also edit POST requests on the **Raw** tab (visible if you select **View** → **Show Request**). The edited request would be similar to the following:

```
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
```

- If WebInspect has detected a SQL injection vulnerability, you can right-click the vulnerable session in 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.



5 To extract all the data from all tables, click **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 box.

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 column values.**

Site Tree		Data Extraction Setti																							
		Status	Details																						
Data for Table...[employee_employee]																									
<table border="1"> <thead> <tr> <th>name</th> <th>id</th> </tr> </thead> <tbody> <tr><td>Shaun Simpson</td><td>6</td></tr> <tr><td>Sam Shober</td><td>5</td></tr> <tr><td>Ray Kelly</td><td>7</td></tr> <tr><td>Raney Eden</td><td>14</td></tr> <tr><td>Nidhi Shah</td><td>2</td></tr> <tr><td>Nick Harbin</td><td>12</td></tr> <tr><td>Matthew Parcell</td><td>1</td></tr> <tr><td>Josh Sweeney</td><td>11</td></tr> <tr><td>John Lyon</td><td>4</td></tr> <tr><td>Joe Sechman</td><td>3</td></tr> </tbody> </table>				name	id	Shaun Simpson	6	Sam Shober	5	Ray Kelly	7	Raney Eden	14	Nidhi Shah	2	Nick Harbin	12	Matthew Parcell	1	Josh Sweeney	11	John Lyon	4	Joe Sechman	3
name	id																								
Shaun Simpson	6																								
Sam Shober	5																								
Ray Kelly	7																								
Raney Eden	14																								
Nidhi Shah	2																								
Nick Harbin	12																								
Matthew Parcell	1																								
Josh Sweeney	11																								
John Lyon	4																								
Joe Sechman	3																								

SQL Injector Tabs

Request Pane

The Request pane contains three 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 two 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 four 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

Follow the steps below to modify the SQL Injector settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **Options**, **Authentication**, or **Proxy** category and enter the settings described in the following sections.
- 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 My Documents\SPi dynamics\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:

_sql_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

Select this option to use a startup macro; then click  to select, edit, or create 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 a type from the **Authentication** list:

Authentication	Description
Automatic	Allow the SQL Injector to determine the correct authentication type. Note that 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. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established.
NT LAN Manager (NTLM)	NTLM is an authentication process used by all members of the Windows NT family of products. It 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.

Authentication Credentials

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

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

Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's Web proxy settings.

Use Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

[Use Firefox proxy settings](#)

Import your proxy server information from Firefox.

[Configure a proxy using a PAC file](#)

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

[Explicitly configure proxy](#)

Configure a proxy by entering 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

[HTTPS Proxy Settings](#)

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

Compliance Manager

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 evaluates each individual response. This session-based scan 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.

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 WebInspect, you can assess your application and then generate a Compliance Report that measures how well your application satisfies this HIPPA 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).

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. For maximum flexibility, you can even create your own agents and associate them with a user-defined requirement.

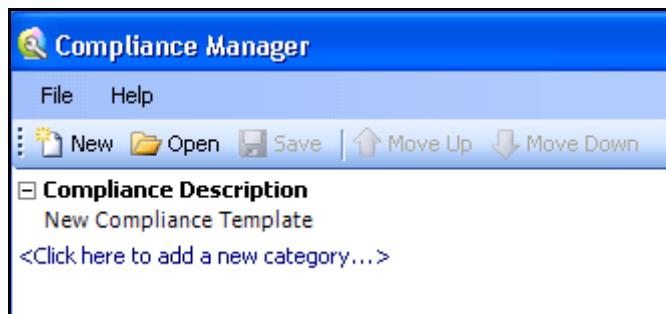
WebInspect includes sample compliance templates that you can edit to fit your company’s specific requirements.

Creating a Compliance Template

Follow the steps below to create a compliance template.

- 1 On the WebInspect menu bar, click **Tools** → **Compliance Manager**.

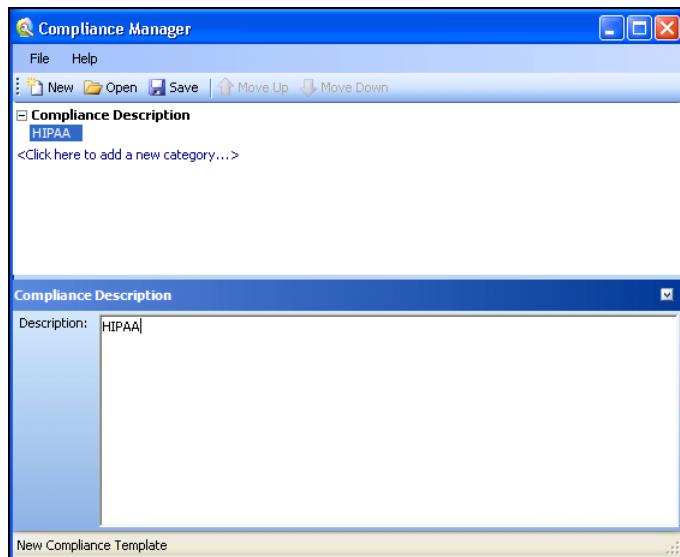
The *Compliance Manager* window opens, displaying the outline of a new template.



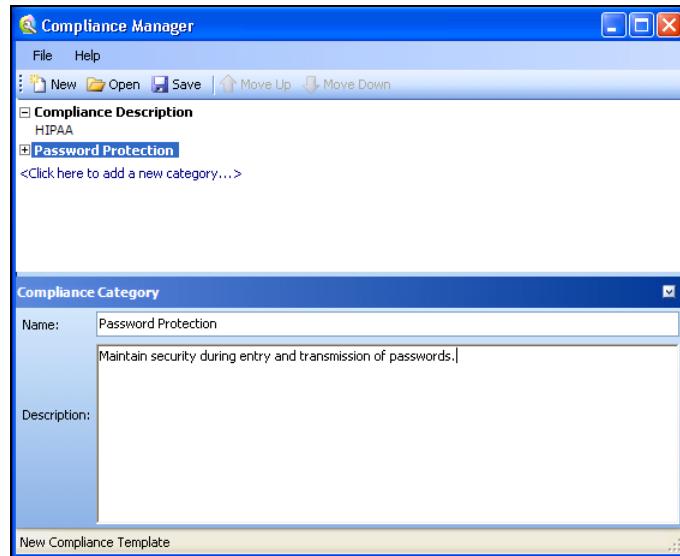
- 2 Click the phrase “New Compliance Template.”

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).



- 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.”



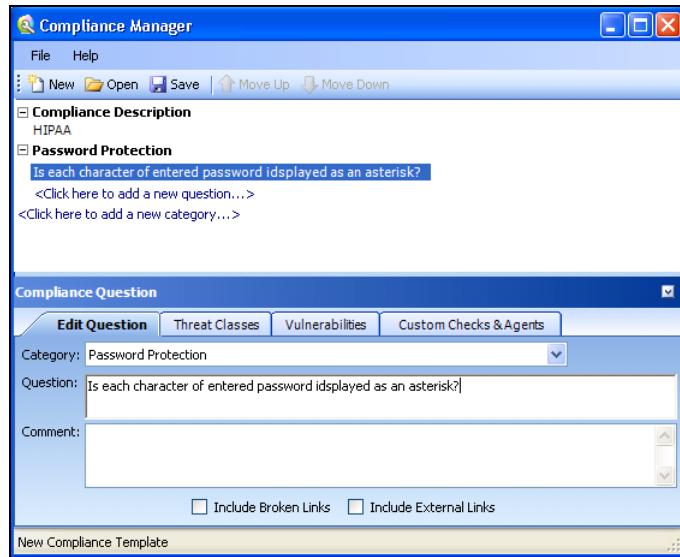
- 6 Click the plus sign to expand the node labeled Password Protection.

- 7 Click the phrase “<Click here to add a new question...>.”

- 8 Click the phase “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?”

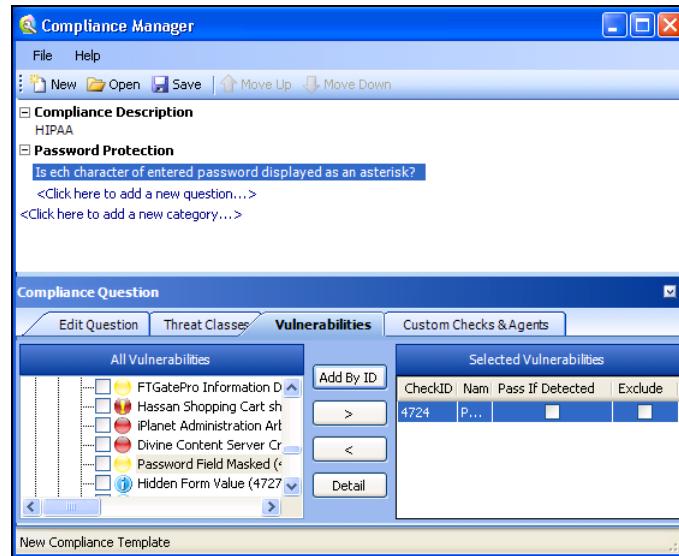


- 10 You can associate this question with threat classes, vulnerabilities defined by HP, or a custom check or agent that you previously created. For this example, click the **Vulnerabilities** tab and then click **Add By ID**.

You can also select a vulnerability and click to include it in the **Selected Vulnerabilities** 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 Not Masked” check.] Note: You can add multiple IDs (one per line).

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



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



Testing for Compliance

Follow the steps below to test your Web site for compliance:

- 1 Create a compliance template.
- 2 Scan your Web site.
- 3 On the WebInspect **Start** page, click **Generate a Report**.
The *Generate a Report* window appears.
- 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 Each Report in a Separate Tab**.
- 9 Select either **Adobe PDF** or **HTML** as the report format.

Adobe Reader 7 or newer 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 WebInspect generates the report and displays it on a tab, you can save a report by clicking the Save Report icon on the toolbar.

Log Viewer

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

Viewing Logs

Follow the steps below to view a log:

- 1 Click **Tools** → **Log Viewer**.

If you open the Log Viewer when a tab containing a scan has the 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 WebInspect's Application settings). They include:
- 5 To locate text within the log, click **Find** on the toolbar or select **Edit** → **Find**.
- 6 To view logs that are not related to a specific scan, click **WebInspect Logs** (on the toolbar).

Web Macro Recorder (Traffic-Mode)

A macro is a recording of the HTTP requests that are generated when you navigate through a Web site or application using the Web Macro Recorder. You can instruct WebInspect to use this recording to enter your Web site and (optionally) navigate through your application.

Any activity you record in a macro will override the scanner settings. For example, if you specify a URL in the Excluded URL setting, and then you actually navigate to that URL when creating a macro, the scanner will ignore the exclusion when it replays the macro.

When starting a Web site assessment with the WebInspect Scan Wizard, you have two opportunities to specify a macro:

- **Workflow-Driven Assessment:** WebInspect audits only those URLs included in the macro that you previously recorded 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 the application. A logout signature is not required. This type of macro is used most often to focus on a particular subsection of the application.
- **Site Authentication:** The macro identifies a log-in page and contains a user name and password that allows you to log on to the target site. The macro must also contain a “logout condition,” which indicates when an inadvertent logout has occurred so WebInspect can rerun this macro to log on again.

When you play a macro, the HP scanner will not send any cookie headers that may have been incorporated in the recorded macro.

WebInspect accommodates three different macro recorders. Use Application Settings - General to specify the one that will be launched by default when creating a macro. See [General Settings](#) on page 195 for more information.

Creating a Macro

Follow the steps below to create a macro:

Task 1: Prepare the Web Macro Recorder

- 1 Close all browsers.
- 2 Start the Web Macro Recorder.
- 3 Click **Edit → Settings** to configure general settings and proxy settings.
- 4 You can exclude the recording of requests containing certain objects by selecting **Filter Rules** from the Macro Recorder's **View** menu. See [Filter Rules](#) on page 329 for more information.

Task 2: Browse the Web Site

- 1 Do one of the following:
 - Select **File → New**.
 - Click the New icon on the toolbar.
 - Click the Record icon.

- 2 Using the browser's Address bar, enter or select a URL.



Note: Internet Explorer version 7 and the .NET Framework are hard-coded not to send requests for localhost through any proxies. Therefore, the Web Macro Recorder will not receive such traffic. This is a documented Microsoft defect. To access a site on "localhost" when using IE7, place a period or dot after "localhost" (for example, http://localhost.:8080/test.html).

- 3 Browse the pages that you want to include in your macro.
- 4 If you want to include a login, be sure to navigate to a page that requires Web form authentication. Then enter a valid user name and password, and submit the data (usually by clicking a button such as **Log On**, **Go**, **Submit**, etc.).
- 5 When finished, close the browser.



If recording a login macro, do not log out before closing.

Task 3: Finish the Macro

- 1 When you close the browser, a dialog box displays the message:

"Are you recording a login macro? (By clicking Yes, auto-detection of the logout condition will be performed.)"

Explanation: When a scanner encounters a hyperlink to another resource, it navigates to that URL and continues its scan. 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 logout occurs, the scanner can either run this macro to log in or request user intervention. In either case, the process hinges on the scanner's ability to recognize when it is no longer logged in.

- Click **Yes** if you want the Web Macro Recorder to analyze the recorded sessions and attempt to detect a "logout" condition.
 - If you do not require a "logout" condition, click **No** and go to Step 6.
 - If you want to specify a condition manually, click **No** and go to Step 3.
 - If your application uses URL rewriting or post data techniques to maintain state within a Web site, click **No**. See **URL Rewriting and Request Parameters** on page 323 for further instructions.
- 2 If the attempt to detect a logout condition is successful, a dialog box displays the following message:

"Would you like to test your login macro?"

 - a To bypass the test, click **No**. Go to Step 3.
 - b To test the macro, click **Yes**.
 - c On the *Test Login Macro* window, the **Address** box contains the URL of a page believed to be viewable only after logging in. If this is, indeed, a "protected" page, click **Go**. Otherwise, enter the URL of a protected page.
 - d Browse to various sections of the site to verify that you are logged in.
 - e Log out and verify that you are prompted to replay the macro.
 - f Click **Done**.

- 3 If the attempt to detect a logout condition is not successful, or if you elected to bypass the auto-detect feature:
 - a On the **Sessions** tab, select a session that you accessed after logging in and click **Detect Logout Condition** (on the toolbar). Do not select the session where you actually logged in.
 - b If the Macro Recorder is unable to determine the logout condition, try selecting other sessions.
 - c If the Macro Recorder is still unable to determine a logout condition, you can manually enter one. Click **Edit Logout Condition** and, on the *Logout Condition Editor* window, select either **Use Regular Expression Extensions** or **Use Text Matching**.
- 4 For a login macro, you may want to delete extraneous sessions (i.e., those not related to or required by the login procedure). To do so, remove the check mark from the unneeded sessions. You should then click **Test Login Macro** to ensure that you retained all necessary sessions.
- 5 Specify which action the scanner should take if it detects that it has logged out of the application. Click either **Play Macro** or **Launch Interactive** (which will allow you to manually log back in).

 Note: If you select **Launch Interactive**, the scanner pauses the scan and presents a dialog allowing you to enter log-in information. This is useful when scanning a site that incorporates a CAPTCHA (i.e., a challenge-response test placed within Web forms to ensure that the response is not generated by a computer). This feature is also used when the Web Macro Recorder is not able to determine a logout condition and the user is not able to define the condition using regular expressions or text matching.
- 6 To save the macro, click **File → Save** (or **Save As**) or click .

Editing the Logout Condition

You can create or edit the criteria used by the Web Macro Recorder to detect a “logged out” condition.

To access the feature, click **Edit Logout Condition**.

If detection of a logout is not required, select **Do no use logout condition**. Otherwise, you can instruct the Web Macro Recorder to use either a regular expression or text matching.

Regular Expression Extensions

If you want the Web Macro Recorder to use a regular expression to detect a logged out condition:

- 1 Select **Use Regular Expression Extensions for a logout signature**.
- 2 Type (or edit) a regular expression that identifies a unique text or phrase that occurs in the server’s HTTP response when a user logs off or when a user who is not logged on requests access to a protected URL.

For example, if your server returns a message such as “Have a nice day” when a user logs off your application, then enter “Have\sa\snice\sday” as the regular expression (“\s” is used in regular expressions to denote a space).

The scanner can also detect that it has logged off 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." In this case, "[STATUSCODE]302 AND [ALL] http://login.myco.com/config/mail?" might be a typical regular expression.

- 3 Click **OK**.

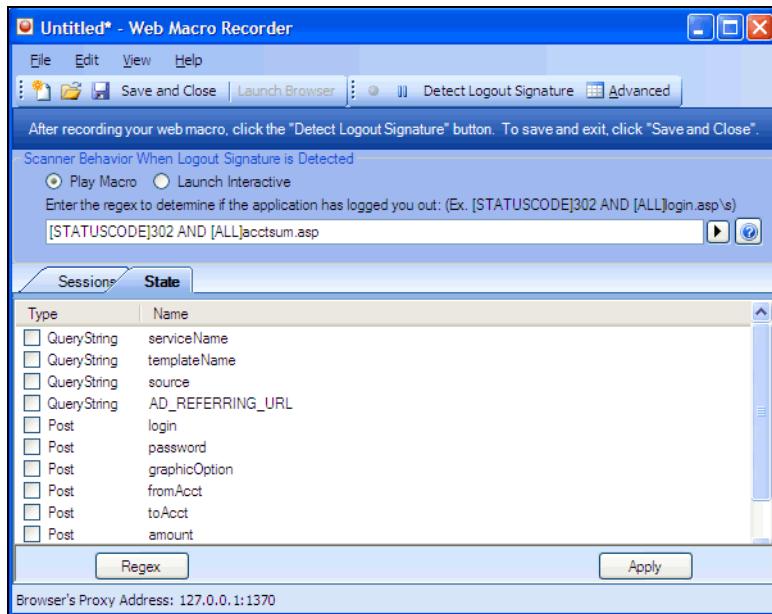
Text Matching

This technique for recognizing "logged out" or "logged in" state assumes that you know that certain text strings will be displayed when either condition occurs. For example, a site may display pages that contain the text "Log In" (usually a hyperlink) whenever a user is not logged in. Similarly, the site may display pages containing text such as "Sign Out," "Log Out," or "Log Off" when the user is logged in.

- 1 Select **Use text matching to determine logged-in state**.
- 2 Under the **Text fragments that indicate logged out state** column, click **Add**.
- 3 In the pop-up window that appears, enter a text string and click **OK**. For example, you might enter "Log In" or "sign in"; note that the search is not case-sensitive.
- 4 Repeat Step 2-3 if additional or alternative text fragments are also present during a "logged out" state.
- 5 In the **Number of text fragments to match** box, enter the number of specified strings that must exist on a page before the Web Macro Recorder considers that page to be in a "logged out" state.
- 6 Under the **Text fragments that indicate logged in state** column, click **Add**.
- 7 In the pop-up window that appears, enter a text string and click **OK**. For example, you might enter "Log Out" or "Sign out."
- 8 Repeat Step 6-7 if additional or alternative text fragments are also present during a "logged in" state.
- 9 In the **Number of text fragments to match** box, enter the number of specified strings that must exist on a page before the Web Macro Recorder considers that page to be in a "logged in" state.
- 10 (Optional) Click **Advanced**.
 - a In the pop-up dialog, enter a URL that should be used to evaluate the state if a page does not contain enough text fragments.
 - b Click **OK**.
- 11 Click **OK**.

URL Rewriting and Request Parameters

If your application uses URL rewriting or request parameters to maintain state within a Web site, select the **State** tab.



You must identify which parameters are used for state management. For example, a PHP4 script can create a constant of the session ID named SID, which is available inside a session. By appending this to the end of a URL, the session ID becomes available to the next page. The actual URL might look something like the following:

.../page7.php?PHPSESSID=4725a759778d1be9bdb668a236f01e01

Because session IDs change with each connection, a recorded macro containing this URL would create an error when you tried to replay it. However, if you identify the parameter (PHPSESSID in this example), then the scanner will replace its assigned value with the new session ID obtained from the server each time the connection is made.

Similarly, some state management techniques use post data to pass information. For example, the HTTP message content may include userid=slbhkelvbk173dhj. In this case, "userid" is the parameter you would identify to the Web Macro Recorder.



Note: You need to identify parameters only when the application uses URL rewriting, posted data or query parameters to manage state. It is typically not necessary when using cookies to manage state. Exception: Delete (uncheck) any cookie that is required for normal operation.

The Web Macro Recorder can identify potential parameters if they occur as posted data or if they exist within the query string of a URL. However, if your application embeds session data in the URL as extended path information, you must provide a regular expression to identify it. In the following example, "1234567" is the session information:

[http://www.onlinestore.com/bikes/\(1234567\)/index.html](http://www.onlinestore.com/bikes/(1234567)/index.html)

The regular expression for identifying the parameter would be:

/\([\\w\\d]+\\)/

- To enter a regular expression, click **Regex** and then use the Regular Expression Editor to create an expression. When you click **OK** (on the regular Expression Editor), the expression is added to the **Type/Name** list.

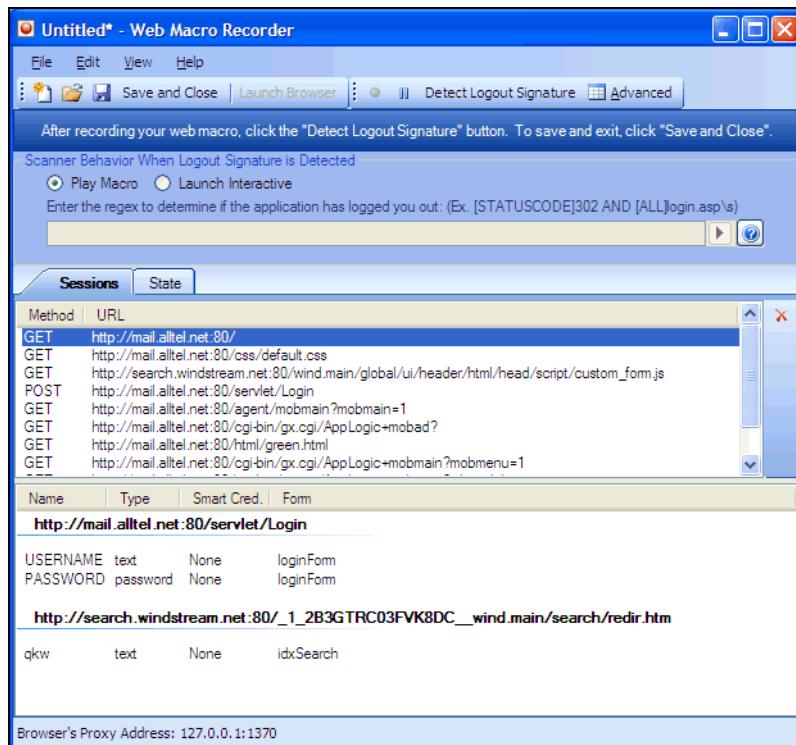
- 2 Select a parameter in the **Type/Name** list.
- 3 Click **Apply**.
- 4 To save the macro, select **File → Save** (or **Save As**)
-or-
click .

Inspecting and Editing a Macro

As you navigate through the target Web site, the Web Macro Recorder transcribes each session, displaying on the **Sessions** tab the method and URL associated with each HTTP request sent to the server.

- 1 Select a session on the **Sessions** tab.

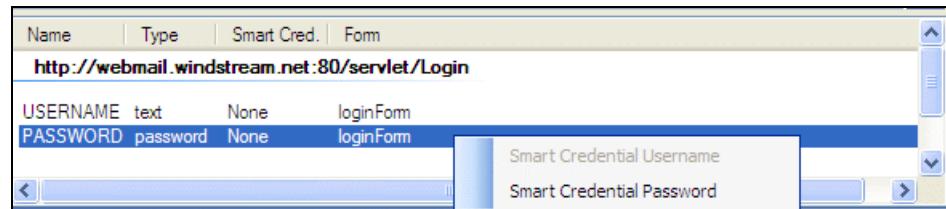
If the associated HTTP response includes “text” or “password” input controls, their name and type are displayed in the lower pane.



In this example, the form and the controls were rendered by the following HTML statements:

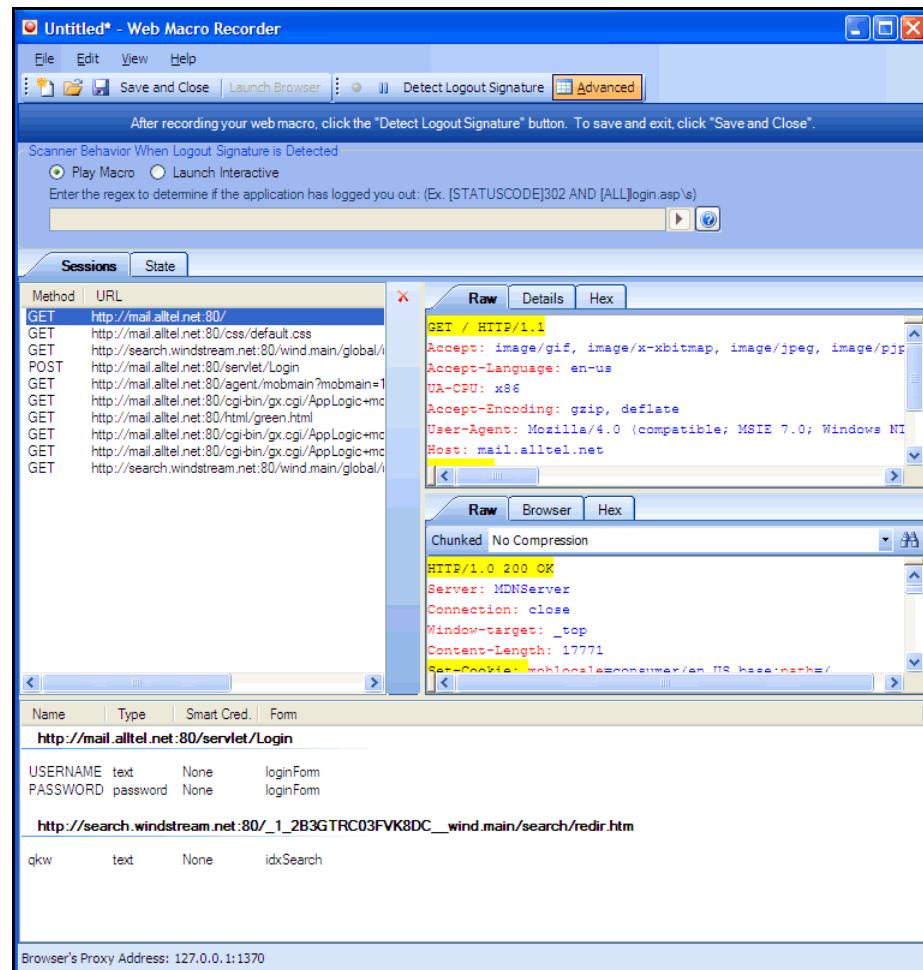
```
<form name="loginForm" action="/servlet/Login" method="POST">
<input type="text" size="16" name="USERNAME" value="">
<input type="password" size="16" name="PASSWORD">
```

- 2 You can designate a control as a “Smart Credential” user name or password. Right-click the control name and select an option from the shortcut menu, as shown below.



If you start a scan using a macro that includes Smart Credentials, then when you scan the page containing the input elements associated with these entries, WebInspect will substitute the password specified in the Authentication options (or, if no user name is specified, the name of the current Windows user). This allows you to create the macro using your own user name and password, yet when someone else runs the scan using this macro, WebInspect will submit that user's name and password.

- 3 If you click the **Advanced** button, the Web Macro Recorder displays the contents of the HTTP request and response in separate panes.



- 4 You can also edit an HTTP request if, for example, you need to change or remove headers, or edit passwords or user names. Simply right-click a session and select **Edit with HTTP Editor** from the shortcut menu to launch the HTTP Editor.

- 5 You can exclude a specific session from the macro by clearing its associated check box, or you can delete a session by selecting the session and clicking the red **X** on the right side of the **Sessions** list (or by right-clicking a session and selecting **Delete Session** from the shortcut menu).

Traffic-Mode Web Macro Recorder Settings

Follow the steps below to modify the Web Macro Recorder settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **General** or **Proxy** category (described below) and enter the settings.
- 3 Click **OK**.

General

Proxy Listener

The Web Macro Recorder 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.

To avoid the possibility of specifying a port that is already in use, select **Automatically Assign Port**.

Save Files in clear text

Select this option if you do not want to save macros in an XML format using Base 64 encoding (which is the default). Saving files in clear text allows you to read the XML tags. The actual data, however, is not rendered in ASCII format and is not human readable.

Keep window always on top

Select this option to keep the Web Macro Recorder displayed on your screen when you switch programs or windows.

Keep params as state only during macro playback

This option affects how the Post and Query parameters in the **State** tab are used. If this setting is off, then the Post and Query parameters that are checked are imported into the scan settings in the **HTTP Parameters Used For State** list. If this setting is on, then the Post and Query parameters that are checked are used as state only during the playback of the macro being recorded.

Automatically follow redirects during playback

If this option is selected, then for any sessions in the macro being recorded that result in a redirect (a 301 or 302 status code, for example), the new redirect will automatically be followed when the macro is played back. The session that is recorded (that is the result of the redirect) will not be played back.

Prompt for credentials when webserver requests authentication

If you select this option, the Web Macro Recorder displays a dialog allowing you to enter a user name and password whenever the server requires authentication to access a site (that is, whenever the server returns a “401 Unauthorized” status).

Note: Certain AJAX, Flash, and ActiveX controls may elicit a 401 status code when authentication, in fact, is not required. You can recognize this situation when the Web Macro Recorder prompts for credentials, but a browser accessing the site does not. For sites where this occurs, this option should not be selected.

Honor only those cookies encountered while recording macros

Problems can sometimes occur when recording a macro on a site that uses persistent cookies, as in the following scenario. When a browser sends its first-ever request to the server, the server sets a cookie and directs this first-time user to a specific resource. However, the next time this browser accesses this server, the browser includes the cookie in the request and, because the client has accessed this site previously, the server directs the client to a different resource. Selecting this option circumvents this behavior.

Disable this option if the site uses JavaScript to set cookies, and delete cookies from your browser before recording the macro.



Tip: If you are unable to log on to a site when using the Web Macro Recorder, but you have no problem logging on when not using the Web Macro Recorder, disabling this option may solve the problem.

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 Macro Recorder should use.

Proxy

Use these settings to access the Web Macro Recorder through a proxy server.

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 Internet Explorer proxy settings

Import your proxy server information from Internet Explorer.

Use Firefox proxy settings

Import your proxy server information from Firefox.

Configure a proxy using a PAC file

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

Explicitly configure proxy

Configure a proxy by entering 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

HTTPS Proxy Settings

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

Web Macro Recorder Menus

The Web Macro Recorder contains the following menus:

File

- **New**—Launch Internet Explorer and begin recording.
- **Open**—Load a previously recorded macro for editing.
- **Save**—Save a macro.
- **Save As**—Save an edited macro under a different file name.
- **Exit**—Close Web Macro Recorder.

Edit

- **Cut**—Delete the selected string and save it to the clipboard.
- **Copy**—Copy the selected string to the clipboard.
- **Paste**—Insert contents of the clipboard.
- **Edit with HTTP Editor**—Open the HTTP Editor and load the selected session.
- **Delete Session**—Remove the selected session from the macro.
- **Start Capture**—Begin recording HTTP requests.
- **Stop Capture**—End recording to HTTP requests.
- **Find**—Specify a string and search for it when using the Advanced view.
- **Settings**—Modify Web Macro Recorder settings.

View

- **Launch Browser**—Open Internet Explorer to navigate through Web site.
- **Test Login Macro**—Open the *Test Login Macro* window to verify creating of a logout condition.
- **HTTP Editor**—Open the HTTP Editor.
- **Toolbars**—View or hide the Detect Logout Condition, Test Login Macro, and Advanced buttons.
- **Filter Rules**—Select a resource type or status code to exclude. For example, sessions where the server response contains an HTTP status code of “404 Object Not Found” are normally not useful. Similarly, sessions that request images are normally not necessary when creating a macro, and simply add clutter to the session list. By selecting **Images** from the Filter Rules list, you avoid the needless recording of sessions such as GET http://www.mywebsite.com:80/services.gif.
- **Advanced**—View or hide panes that display the contents of HTTP requests and responses. Note that when editing a saved macro, pages will not be rendered in the **Browser** tab.

Help

- **Web Macro Recorder Help**—Open the Help file to the default topic.
- **Index**—Open the Help file, displaying the index pane.
- **Search**—Open the Help file, displaying the search pane.
- **About Web Macro Recorder**—Open a window that displays information about the Web Macro Recorder.

Web Macro Recorder (Event-Based IE Compatible)

A macro is a recording of the events that occur when you access and log in to a Web site using the Event-Based Web Macro Recorder. You can subsequently instruct the HP scanner (WebInspect or QAInspect) to begin a scan using this recording.

A login macro should contain events recorded during a login procedure and incorporates logic that will prevent WebInspect from terminating prematurely if it inadvertently logs out of your application. When scanning a site, WebInspect analyzes every server response to determine the state. If the scanner determines at any time that it is logged out, it runs this macro to log in, and then resumes crawling or auditing the site at the point where the logout occurred. When beginning a scan through the WebInspect scan wizard, you can specify a login macro at Step 2 under Site Authentication.

You can access the event-based Web Macro Recorder in several ways:

- When starting a site scan, select Site Authentication (on Step 2 of the WebInspect Scan Wizard) and click **Record**.
- On the WebInspect toolbar, click **Tools** → **Web Macro Recorder**.
- In default scan settings, click **Scan Settings** → **Authentication** → **Use a login macro**.
- From the Windows Start menu: Click **Start** → **HP** → **HP Security Toolkit** → **Web Macro Recorder**.

Note: WebInspect accommodates three different macro recorders. Use Application Settings - General to specify the one that will be launched by default when creating a macro. See [General Settings](#) on page 195 for more information.

Recording a Log-In Macro

After opening the Web Macro Recorder, use the following procedure to record a log-in macro.

- 1 Select **File** → **New** → **Login Macro**.
- 2 Click **Record**.
- 3 In the **Address** box, enter the URL of the target Web site and click  (or press **Enter**).
The Web Macro Recorder renders the resource like a browser and records each event on the Events tab in the dockable pane positioned (by default) at the bottom of the window.
- 4 If necessary, navigate to the login screen.
- 5 Enter a valid user name and password, and submit the credentials (usually by clicking a button such as Log On, Go, Submit, etc.).
- 6 Click **Stop** (to the right of the Address bar) or **Stop Recording** (on the Status bar).
- 7 When prompted to play your macro, click **OK**.

The macro plays by sequentially executing each enabled event listed on the Events tab. A message prompts you to either confirm the success of the macro and specify a logout condition or (assuming that the macro was not successful) troubleshoot the macro.

- 8 Do one of the following:
 - To specify a logout condition, select **Yes** and click **Finished**. Go to [Specifying a Logout Condition](#) (below).
 - To troubleshoot, select **No** and click **Next**. Go to [Troubleshooting a Macro](#) on page 331.

Specifying a Logout Condition

- 1 Navigate to a page where you are logged out (usually by clicking a button such as **Log Out**, **Log Off**, or **Exit**).
- 2 Do one of the following:
 - If the browser always displays this page when you log out, click **This page displays when I have logged out** (on the Selection Mode bar that appears directly under the Web Macro Recorder toolbar).
 - If the browser displays a page that contains an element or control that appears only when you are logged out, click **Select Logout Indication** (on the Selection Mode bar) and then click the element or control. For example, if a Login button appears when you have logged out, click **Select Logout Indication** and then click the Login button. Your selection appears on the **Logout Elements** tab.
 - If you want the scanner to search each page for a condition that matches a regular expression that you create, click **Add logout regex**. See [Regular Expression Editor](#) on page 267 for details.
- 3 Select **File → Save (or Save As)**.

Note: You can specify a logout condition at any time by clicking **Actions → Add Logout Condition**.

Specifying a Confirmation Element

After creating the macro, you may optionally identify a “confirmation element” that indicates that you have logged in successfully. This is particularly useful for those sites that, following a successful login, display a specific element or control on every page. Some sites, for example, always present a “Log Out” button after the user has logged in. Identifying this confirmation element increases the probability that WebInspect will be able to recognize the “logged in” condition.

Once you identify a confirmation element, if the scanner does not detect that element on the page, it assumes the macro has failed and will attempt to replay the macro up to three times. If the confirmation hint is not detected during one of these playbacks, the scanner produces an error and stops trying to use the macro.

- 1 Navigate to a page that appears after you log in.
- 2 Click **Actions → Add Confirmation Element**.
- 3 Do one of the following:
 - If this page always appears after you log in, select **This page displays when I have logged in**.
 - Click **Select Confirmation Element** and then click an element on the page that appears only when you are logged in.

Troubleshooting a Macro

When troubleshooting your recorded macro, you have the following choices:

- **Replay Macro** - Try this solution first. The Web Macro Recorder normally plays the macro at the fastest possible speed, which may compromise performance. Use the slider to select either **Fast** (which is half the speed at which the macro was recorded) or **Original** (which mimics the speed at which the macro was originally recorded).

- **Switch to Traffic Mode** - This closes the Event-Based Web Macro Recorder and opens an alternate web macro recorder that attempts to create macros by analyzing the http traffic. This tool was included with WebInspect version 8.1 and earlier.
- **Adjust macro hints** - Allows you to add or change confirmation elements and/or logout conditions.
- **Re-record Macro** - This choice deletes all data and returns you to the beginning point, where you can try again to create a successful macro.

Editing a macro

After recording a macro, you can modify its contents by excluding certain events.

For example, if you entered the wrong validation credentials while attempting to log in, and then entered the correct credentials, you can remove the erroneous log-in events simply by clearing the check box (in the Include column of the **Events** tab) next to the event you want to exclude.

Events		
Include	Type	Info
<input checked="" type="checkbox"/>	SetValue	Set the value of the [uid] element to [dthackery@windstream.net]
<input checked="" type="checkbox"/>	SelectElement	Select the [password] element
<input checked="" type="checkbox"/>	SetValue	Set the value of the [password] element to [*****]
<input checked="" type="checkbox"/>	SelectElement	Select the [Submit] element

Ordinarily, the best practice is to re-record the macro instead of editing it. However, for an extremely lengthy or complex macro, you can first attempt to modify it. Excluded events are not actually removed until you save the macro, so be sure to test the modified macro (by playing it) before you save it.

You might also need to add events for those situations where events are not recorded (such as login elements located in an I-frame).

The Web Macro Recorder events are defined in the following table.

Event	Definition
WaitForPageLoad	Wait for the browser to complete the processing of pages.
NavigateTo	Navigate to the specified URL.
WaitForElement	Wait for element to be rendered on current page. This is used most often to render cascading menus.
WaitNumberOfSeconds	Pause for a specific number of seconds.
Click	Simulate a mouse click on an element.
MouseUp	Simulate any mouse button being released over an element.
MouseDown	Simulate any mouse button being pressed while the pointer is over an element.
SetValue	Simulate entering a value associated with an element.
JavaScript	Execute JavaScript.

Example: Adding Elements for I-Frame Login

The most frequently encountered failure to record a login macro occurs when the login elements are contained within an iframe. During recording, you might enter a user name and password, and then click the Signin button, but nothing occurs when you play the macro.

You can edit the recorded events or you can begin by recording a new macro. If you edit the recording:

- 1 Click **Stop** (on the Status bar).
- 2 Deselect (remove the check marks next to) those events that occur after the page is loaded.

Create an event for the user name element

- 1 Right-click the WaitForPageLoad event and select **Insert new event here** from the shortcut menu.
- 2 On the *Event Properties* dialog, click the drop-down arrow on the **Type** list and select **Click**.
- 3 Click **Locate an Element** (at the bottom of the dialog).
- 4 Move the mouse pointer to and click on the user name element (which may be labeled "name," "user," "e-mail address" or other such identifier).
- 5 When the *Event Properties* dialog displays information about the element, click **OK**.

Note that the event is added after (following) the event on which you clicked.

Add a value to the user name element

- 1 Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
- 2 On the *Event Properties* dialog, select **SetValue** from the **Type** list.
- 3 Click **Locate an Element** (at the bottom of the dialog).
- 4 Click the user name element.
- 5 On the *Event Properties* dialog, enter a user name in the **Value** box and click **OK**.

Create an event for the password element

- 1 Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
- 2 On the *Event Properties* dialog, select **Click** from the **Type** list.
- 3 Click **Locate an Element** (at the bottom of the dialog).
- 4 Click the password element.
- 5 When the *Event Properties* dialog displays information about the element, click **OK**.

Add a value to the password element

- 1 Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
- 2 On the *Event Properties* dialog, select **SetValue** from the **Type** list.
- 3 Click **Locate an Element** (at the bottom of the dialog).

- 4 Click the password element.
- 5 On the *Event Properties* dialog, enter a password in the **Value** box and click **OK**.

Submit the user name and password

- 1 Right-click the event that you just added and select **Insert new event here** from the shortcut menu.
- 2 On the *Event Properties* dialog, select **Click** from the **Type** list.
- 3 Click **Locate an Element** (at the bottom of the dialog).
- 4 Click the submit element (which may be labeled “Submit,” “Sign In,” or other such identifier).
- 5 On the *Event Properties* dialog, click **OK**.

Dynamic 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). The simplest example is where the challenge asks for a password and the valid response is the correct password.

Many Web sites now present multiple challenges to the user. Typically, when a user first registers with a Web site, the site presents a list of questions to which the user provides answers. For example:

What is your favorite color?
What was the name of your first pet?
In what town or city were you born?
What was the make of your first automobile?

When the user later attempts to log in, the Web site presents two or more of these challenges. Some sites also create groups of challenges, and dynamically present questions from different groups on each subsequent log-in attempt, as demonstrated in the following example.

When registering for the following example Web site, the user is asked to provide answers to nine questions, which are arranged into three groups of three questions each, as follows.

Group 1

“What is your name?”, “Smith”
“What is your favorite color?”, “blue”
“What is the name of your first grade teacher?”, “Williams”

Group 2

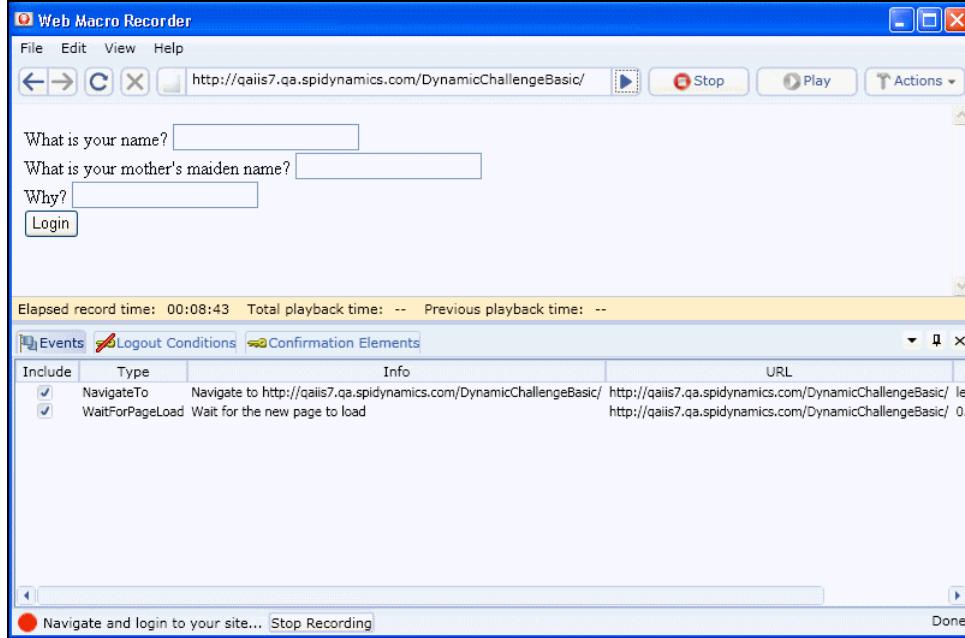
“What is your mother's maiden name?”, “Larrimore”
“In what state were you born?”, “Delaware”
“What is the name of your favorite pet?”, “Rusty”

Group 3

“Why?”, “Albatross”
“What is your paternal grandmother's first name?”, “Esther”
“What is the capital of the state you live in?”, “Atlanta”

In this example, the application randomly selects a number between 1 and 3 (inclusive) and then displays the corresponding ordinal question (first, second, or third) from each group.

- 1 Start the Web Macro Recorder, click **Record**, and enter the URL of the log-in page.



The source code for the pertinent area of the form is:

```
<label for="Q1"> What is your name?</Label><input id="Q1" name="Q1" /> <br>
<label for="Q2"> What is your mother's maiden name?</Label><input id="Q2" name="Q2" /> <br>
<label for="Q3"> Why?</Label><input id="Q3" name="Q3" /> <br><input type="submit" value="Login" />
```

This illustrates that the label for each question is Q1, Q2, and Q3; similarly, the ID and name for each text box into which the user enters the response is Q1, Q2, and Q3.

- 2 On the log-in page, enter a value for each input element and click **Login**.
- 3 Assuming that you logged in correctly, click **Stop**.
- 4 When prompted to play your macro, click **Cancel**.

To modify the macro so that it accommodates a random presentation of authentication questions:

- 1 Navigate to the log-in page.
- 2 Click the **Events** tab.
- 3 Right-click the first SetValue element and choose Select security question for this element.
 - a Click **Select Security Question** (just below the toolbar).
 - b Click on the label for the first security question (in this example, "What is your name?").
The *Question-Answer Groups* dialog appears.
 - c In this example, we know that the first question is a member of the Q1 group. So click the **Add** button, enter "Q1" in the Group Name box, and click **OK**.

Note: If your program does not divide questions and answers into groups, but presents the same set of questions at each log-in attempt, ignore the Group Name controls.

- d Click **Click here to add new question/answer pair**.
 - e Enter the first question and answer pair. In this example:
Question: What is your name?
Answer: Snouck
 - f Repeat Steps 3d-3e, entering the second and third question/answer pair in Group 1.
 - g Click **OK**.
Note that a **Sec. Questions** column is added to the **Events** tab.
 - h Click the drop-down arrow for this element in the **Sec. Questions** column on the **Events** tab and select Q1.
- 4 Right-click the second SetValue element and choose **Select security question for this element**.
- a Click **Select Security Question** (just below the toolbar).
 - b Click on the label for the second security question (in this example, “What is your mother’s maiden name?”).
 - c Click the drop-down arrow for this element in the **Sec. Questions** column on the **Events** tab and select Manage.
 - d On the *Question-Answer Groups* dialog, click **Add**, enter a group name of “Q2” and click **OK**.
 - e Add the three security question/answer pairs for the Q2 group, following the procedure outlined in Step 3.
- 5 Right-click the third SetValue element and choose **Select security question for this element**.
- a Click **Select Security Question** (just below the toolbar).
 - b Click on the label for the third security question (in this example, “Why?”).
 - c Click the drop-down arrow for this element in the **Sec. Questions** column on the **Events** tab and select Manage.
 - d On the *Question-Answer Groups* dialog, click **Add**, enter a group name of “Q3” and click **OK**.
 - e Add the three security question/answer pairs for the Q3 group, following the procedure outlined in Step 3.
- 6 Click **Play** to test the macro.

When troubleshooting the macro, it is usually helpful to right-click an entry on the **Events** tab and select **Playback macro to this event**.

Logout Elements

When the *Playback Successful?* dialog appears, the first of three messages at the bottom of the dialog pertains to logout conditions. These are elements, pages, or regular expressions that indicate to the Web Macro Recorder (and the scanner) that the user is no longer logged in to the site or application.

If the message is “Logout conditions have been specified for this macro,” the Web Macro Recorder has recognized the logout condition you specified.

However, if the message is “Unable to auto-detect logout conditions,” then either:

- You did not instruct the Web Macro Recorder to automatically detect logout elements (see **Settings**).
- The Web Macro Recorder was unable to auto-detect elements.
- You did not manually specify a logout condition.

To correct this defect, click **Edit** → **Settings** → **Auto-Detection** and do one of the following:

- Select **Auto-detect logout conditions** and choose one or more of the standard logout elements (or create a custom logout element).
- Clear **Auto-detect logout conditions**, click **OK** to save the settings, and then:
 - a Click and select **Add Logout Condition**.
 - b Use the Forward and Back buttons to navigate to a page that contains a logout element.
 - c Do one of the following:
 - Click and then click the page element that appears only when you are in a “logged out” condition.
 - If the entire page appears only after the user has logged out, click .
 - If you want the scanner to search each page for a logout condition that matches a regular expression that you create, click .

To delete a logout condition from the macro, click the **Logout Conditions** tab (in the Web Macro Recorder's lower pane), right-click a condition, and select **Delete**.

Using a Regular Expression for Logout Detection

If you want the scanner (and Web Macro Recorder) to use a regular expression to detect a logged out condition:

- 1 Select **Add Logout Regex**.

The Regular Expression Editor opens.

- 2 Enter a regular expression that identifies a unique text or phrase that occurs in the server's HTTP response when a user logs off or when a user who is not logged on requests access to a protected URL.

For example, if your server returns a message such as “Have a nice day” when a user logs off your application, then enter “Have\s\snice\sday” as the regular expression (“\s” is used in regular expressions to denote a space).

The scanner can also detect that it has logged off 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.” In this case, “[STATUSCODE]302 AND [ALL] http://login.myco.com/config/mail?” might be a typical regular expression. See [Regular Expression Extensions](#) for more information.

- 3 Click **OK**.

Confirmation Elements (Hints)

When the *Playback Successful?* dialog appears, the second of three messages at the bottom of the dialog pertains to confirmation elements. These are elements or pages that indicate to the Web Macro Recorder (and the scanner) that the user is logged in to the site or application.

If the message is “Confirmation elements have been specified for this macro,” the Web Macro Recorder has recognized the element that you specified as indicating that the user is logged in.

However, if the message is “Unable to auto-detect confirmation conditions,” then either:

- You did not instruct the Web Macro Recorder to automatically detect confirmation elements (see **Settings**).
- You instructed the Web Macro Recorder to automatically detect confirmation elements, but the Web Macro Recorder could not recognize the element you specified (or you failed to specify an element).
- You did not manually specify a confirmation element.

To correct this defect, click **Edit** → **Settings** → **Auto-Detection** and do one of the following:

- Select **Auto-detect confirmation** conditions and choose one or more of the standard elements (or create a custom element).
- Clear **Auto-detect confirmation** conditions, click **OK** to save the settings, and then:
 - a Click  and select **Add Confirmation Element**.
 - b Use the Forward and Back buttons  to navigate to a page that contains a confirmation element.
 - c Do one of the following:
 - Click  and then click the page element that appears only when you are in a “logged in” condition.
 - If the entire page appears only after the user has logged in, click .

Unsupported Elements

While recording your macro, the Web Macro Recorder displays a warning if you click an unsupported element. These non-HTML elements include objects created using the following technologies:

- Applets
- ActiveX
- Silverlight
- Flash
- Cross-Domain Iframes

If these objects are not required components of your macro, there is no problem. The Web Macro Recorder simply ignores the object and continues to record events as you generate them by navigating through the site.

However, if an unsupported element contains an essential component (such as a login form), the macro will not succeed.

You might avoid this issue by switching to the traffic-mode Web Macro Recorder.

Event-Based Web Macro Recorder Settings

Follow the steps below to modify the Web Macro Recorder settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **Application** or **Macro** category (described below) and enter the settings.
- 3 Click **OK**.

Application Settings

General

Show startup window

The startup window appears when the Web Macro Recorder is launched and displays a shortcut menu that allows you to begin creating or editing a login macro.



Compress macro files

Applies a compression algorithm to reduce the size of the saved macro.

Encrypt macro file

Applies an encryption algorithm to the saved macro to provide security.

Network Authentication Credentials

If network authentication is required, provide a user name and password that will allow access to the network.

Troubleshooting

Highlight failed events

If you select this option, the program displays failed events with a background color.

- Red highlight: The macro event caused the macro to fail.
- Orange highlight: The event failed, but playback continued.

Ignore events after final page load

In most cases, the events that occur after loading the final page in the macro are not significant and do not affect the playback of the macro.

Auto-Detection

During the recording process, you can manually specify a logout element (an object that appears on the page to indicate that you have logged in successfully) and a confirmation element. If auto-detection is enabled and the program automatically detects a logout element during the recording process, the wizard that appears once playback is complete will reflect this and you will not be prompted to select a logout element.

To instruct the Web Macro Recorder to automatically detect elements, select **Auto-detect logout elements** and/or **Autodetect confirmation hints**.

To identify which of the standard elements will trigger automatic detection, select or clear the associated check box next to the element in the Standard list.

To create a custom element:

- 1 Click **Add**.
- 2 In the **Value** box, enter a text string that appears somewhere within the page.
- 3 Click **OK**.
The element appears in the Custom list.
- 4 In the **Type** column, click the down arrow and select the element type: **Confirmation** or **Logout**.

Proxy

If you need to use a proxy server to access the target Web site:

- 1 Select **Use Proxy**.
- 2 Enter the IP address or host name of the server.
- 3 Enter the server's port number.

Macro Settings

General

Smart Credentials

If you start a scan using a macro that includes Smart Credentials, then when you scan the page containing the input elements associated with these entries, WebInspect will substitute the user name and password specified here. This allows you to create the macro using your own user name and password, yet when other persons run the scan using this macro, they can substitute their own user name and password.

To enable this feature, you must first record a macro and then associate one SetValue event in the Events grid as a user name and another SetValue event as a password.

Replacement URL

If you select **Enable URL Replacement**, the host name entered as the Start URL in the Scan Wizard will be dynamically inserted into each URL for this macro. For example, suppose you record a macro for www.testsite.com. At a later point in time, www.testsite.com is renamed to www.testsite2.com. Instead of recording an entirely new macro, you could reuse the original one and enable URL replacement.

IE Dialogs

Microsoft's Internet Explorer may sometimes display dialogs that are not related to the actual content of the Web page. For example, the browser's security feature may present a modal dialog with the following message: "Do you want to view only the webpage content that was delivered securely?" If this occurs during playback of a macro, the scanner will halt until the user presses **Yes** or **No**. You can avoid this interruption by selecting **Use IE Dialog Suppression**.

Several conditions are defined by default. You may, however, define a condition that meets your specific requirements. To do so:

- 1** Click **Add**.
- 2** Enter the requested information.
 - Dialog Caption: Enter the text that appears on the title bar of the dialog box.
 - Dialog Text: Enter the text that appears as the message content.
 - Button: Enter the text that appears on the button that the macro should automatically "press."

The utility that performs this check is case-sensitive, so be sure to enter the text string exactly as it appears.

- 3** Click **OK**.

Web Macro Recorder (TruClient)

A macro is a recording of the activity that occurs when you navigate through a Web site or application using the Web Macro Recorder. You can instruct WebInspect to use this recording to enter your Web site and (optionally) navigate through your application.

A login macro should contain events recorded during a login procedure and incorporates logic that will prevent WebInspect from terminating prematurely if it inadvertently logs out of your application. When scanning a site, WebInspect analyzes every server response to determine the state. If the scanner determines at any time that it is logged out, it runs this macro to log in, and then resumes crawling or auditing the site at the point where the logout occurred. When beginning a scan through the WebInspect scan wizard, you can specify a login macro at Step 2 under Site Authentication.

You can access the TruClient Web Macro Recorder in several ways:

- When starting a site scan, select Site Authentication (on Step 2 of the WebInspect Scan Wizard) and click **Record**.
- On the WebInspect toolbar, click **Tools → Web Macro Recorder**.
- In default scan settings, click **Scan Settings → Authentication → Use a login macro**.
- From the Windows Start menu: click **Start → HP → HP Security Toolkit → Web Macro Recorder**.

Note: WebInspect accommodates three different macro recorders. Use Application Settings - General to specify the one that will be launched by default when creating a macro. See [General Settings](#) on page 195 for more information.

This Web Macro Recorder is an adaptation of the Ajax TruClient technology originally developed for use with HP LoadRunner and HP Performance Center. It does not incorporate or support all the capabilities of the fully-featured version.

The TruClient Macro Recorder does not support the recording of Flash or Silverlight applications.



Note: When launching the TruClient 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.

Recording a Macro

This task describes the basic steps involved in interactively recording a login macro.

Task 1: Record the login

- 1 Enter the URL of the target site in the address bar at the top of the window and press Enter.
- 2 If necessary, navigate to the login page.

Note: If the browser displays a message that the connection is untrusted, click **I Understand the Risks** and then click **Add Exception** before continuing to next step.

- 3 Click **Record**. All of your actions will be recorded and displayed in the pane on the left. You can pause or stop the script and continue recording from any point in the script.
- 4 Enter your user name and password.
- 5 Submit the login credentials by clicking the appropriate button (such as Login, Log On, Submit, Enter, etc.).
- 6 Click **Stop**.

Task 2: [Replay the macro](#)

Replay the macro, correcting any errors that occur during the process.

You can use the **Play** button in the left pane  or the **Play** button in the right pane  . If you experience errors, see [Debugging Macros](#) on page 350.

Task 3: [Identify a “logged out” condition](#)

Navigate to a protected page (that is, one that you cannot access without being logged in to the application) and click  . TruClient will attempt to detect a logout condition.

- If TruClient is able to automatically detect a logout condition, your macro has been created.

For an automatic logout condition, TruClient attempts to determine if your site employs an HTTP redirect to a login page when login state has been lost. If so, a regular expression logout condition will be generated automatically based on the ‘Location’ header of the redirect. This automatically generated regular expression could change at scan time depending on what navigation parameters are specified in the scan settings. Navigation parameters are needed to help uniquely identify the URL referenced in the ‘Location’ header of the redirect. Because this regular expression may change during the scan, it has not be made available in the interface.

- If automatic logout condition detection was not successful, click  and then, on the page that is being displayed, click an element or control that appears only when you are logged out. For example, if a Login button appears when you have logged out, click **Select** and then click the Login button.

Note: If you prefer, you may elect to identify a specific URL that always appears after the user logs out, or you can specify a regular expression that describes a resource that appears after logging out. To do so, finish this step and then click  and select **View Logout** (or click  in the left pane).

Task 4: [Modify the logout or enter parameters](#)

You have completed the macro. The next steps are optional.

- To examine or modify the logout condition (to identify a specific URL that always appears after the user logs out or to specify a regular expression that describes a resource that appears after logging out), click  and select **View Logout**.
- To parameterize the login credentials, click  and select **Parameterize Input**.

Task 5: Save the macro

Click **Save**  to save the macro.

Parameterizing Input

When recording a log-in macro, you can use the Parameters Editor for two different features:

- Create a parameter for the user name and password, allowing testers to use their own authentication credentials when starting a scan.
- Create a parameter for the URL, allowing testers to designate an alternate URL when the macro is run. 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 selecting Site Authentication on Step 2 of the Scan Wizard.

Procedures for creating these parameters are detailed below.

Using Name and Password Parameters

Task 1: Create Parameters

- 1 After creating and testing your log-in macro, click **Open Parameters Editor** (or, if the instruction at the top of the right pane is “Your macro is now complete,” click **Options** and select **Parameterize Input**).

The Parameters Editor opens.

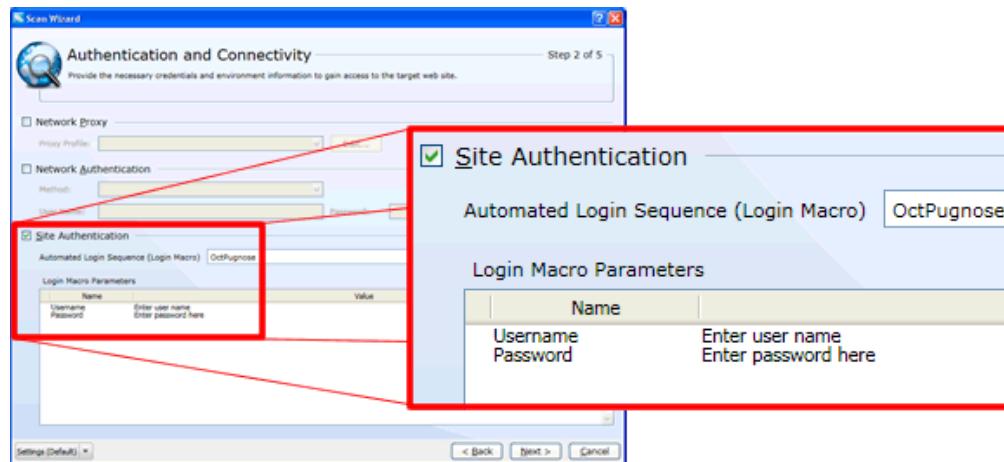
- 2 Click  to add a parameter.
- 3 In the **Name** box, enter a name for the parameter (such as “Username”).
- 4 In the **Value** box, enter the label that you want to appear on the Login Macro Parameters grid (such as “Enter user name”).
- 5 Click **Apply**.
- 6 Click  to add a second parameter.
- 7 In the **Name** box, enter a name for the parameter (such as “Password”).
- 8 In the **Value** box, enter the label that you want to appear on the Login Macro Parameters grid (such as “Enter password here”).
- 9 Select **Encrypted** if the value should be encrypted before transmission to the Web server.
- 10 If you renamed the parameter, click **Apply**.
- 11 Click **Close**.

Task 2: Assign Parameters to Steps

- 1 Select the macro step (in the left pane) that contains the user name.
- 2 Click the drop-down arrow on the far right to open the Step Editor.
- 3 Click **Arguments**.

- 4 Highlight the entire contents of the **Value** box, right-click the highlighted text, and select **Replace with a Parameter**.
- 5 On the *Enter Parameter Name* dialog, select the parameter (“Username” in this example) from the **Select Parameter** list and click **OK**.
- 6 Select the macro step that contains the password.
- 7 Click the drop-down arrow on the far right to open the Step Editor.
- 8 Click **Arguments**.
- 9 Highlight the entire contents of the **Value** box, right-click the highlighted text, and select **Replace with a Parameter**.
- 10 On the *Enter Parameter Name* dialog, select the parameter (“Password” in this example) from the **Select Parameter** list and click **OK**.
- 11 Save the macro.

When you start the scan and select this log-in macro on Step 2 of the Scan Wizard, the parameters appear in the Login Macro Parameters grid (illustrated below). The tester simply replaces the parameters with a valid user name and password.



Using URL Parameters

Task 1: Create Parameter

- 1 After creating and testing your log-in macro, click **Open Parameters Editor** (or, if the instruction at the top of the right pane is “Your macro is now complete,” click **Options** and select **Parameterize Input**).

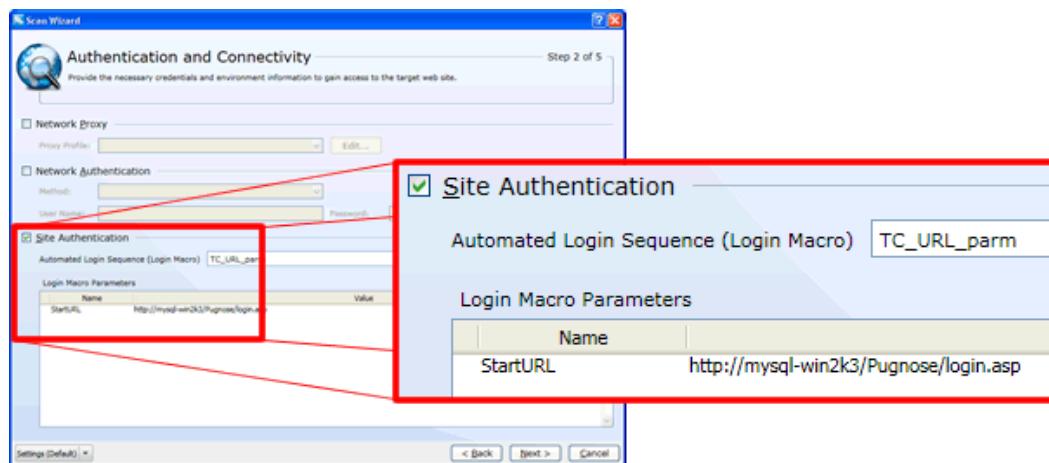
The Parameters Editor opens.

- 2 Click to add a parameter.
- 3 In the **Name** box, enter a name for the parameter (such as “StartURL”).
- 4 In the **Value** box, enter the actual Host Name or URL (such as www.testsite.com).
- 5 Click **Apply**.
- 6 Click **Close**.

Task 2: Assign Parameters to Steps

- 1 Select the macro step (in the left pane) that contains the URL (“Navigate to...”).
- 2 Click the drop-down arrow on the far right to open the Step Editor.
- 3 Click **Arguments**.
- 4 Highlight the entire contents of the **Value** box, right-click the highlighted text, and select **Replace with a Parameter**.
- 5 On the *Enter Parameter Name* dialog, select the parameter (“StartURL” in this example) from the **Select Parameter** list and click **OK**.
- 6 Save the macro.

When you start the scan and select this log-in macro on Step 2 of the Scan Wizard, the parameter appears in the Login Macro Parameters grid (illustrated below). The tester either leaves the parameter unchanged (to access the original URL) or enters the URL of the new site.



Recording a Multi-Challenge Macro

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). The simplest example is where the challenge asks for a password and the valid response is the correct password.

Many Web sites now present multiple challenges to the user. Typically, when a user first registers with a Web site, 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 you born?
What was the make of your first automobile?

When the user later attempts to log in, the Web site presents two or more of these challenges.

Some sites also create groups of challenges, and present questions from different groups on each subsequent log-in attempt, as demonstrated in the following example.

When registering for the sample Web site, 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: red

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?

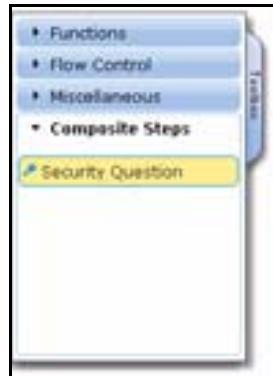
Login

When creating a macro for a challenge/response type of login, you must know all possible question-and-answer combinations, even if only a portion or subset of those combinations may be presented during any one episode. You will enter these combinations manually, before recording the login.

Use the following procedure to create a login macro for this hypothetical web page.

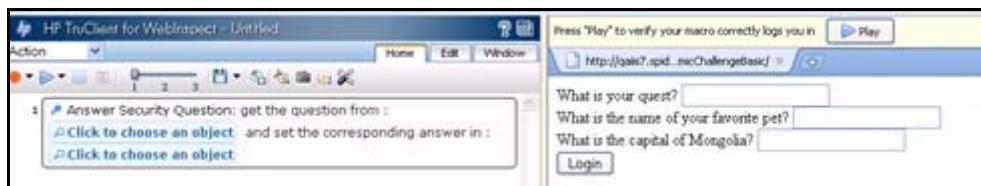
For each of the three question-and-answer locations:

- 1 Click the Toolbox (represented by a vertical tab on the left side of the left pane).
- 2 Click **Composite Steps**.

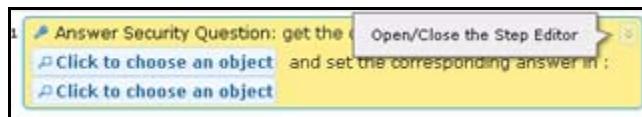


- 3 Drag the Security Question element and drop it in the left pane.

- Click the first “Click to choose an object” button and then, in the right pane, click the object representing the first question (usually a label).

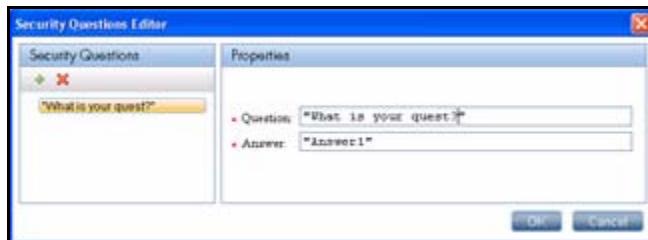


- Click the second “Click to choose an object” button and then, in the right pane, click the object representing the answer (usually a text box).
- In the left pane, open the Step Editor for the step you just created (by clicking in the upper right corner of the step).



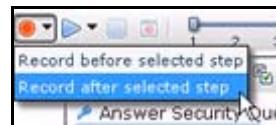
7 Click Arguments.

- Click to create a security question.
- Using the Security Questions Editor, click the plus sign to add a new question.
- In the **Question** box, replace the default text "Question1" 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.



- In the **Answer** box, enter the correct response.
- Click **OK**.
- Repeat steps 8-12 to add the information for the second question that may appear in the first location (in this example, "What is the name of your favorite pet?").
- Repeat steps 8-12 to add the information for the third question that may appear in the first location (in this example, "What is the capital of Mongolia?").
- Refresh the page: click in the right pane and press F5 (or right-click and select **Reload**) until the second set of questions appears.
- Repeat Steps 1-14 to add a second step that contains three different questions and answers.
- Refresh the page: click in the right pane and press F5 (or right-click and select **Reload**) until the third set of questions appears.
- Repeat Steps 1-14 to add a third step that contains three different questions and answers.

- 19 After creating steps for all possible question-and-answer combinations, select the last step.
- 20 Click the drop-down arrow on the Record button and select **Record after selected step**.



- 21 On the web page, click the login control (the button or hyperlink labeled Log In, Log On, Submit, etc.).
- 22 Click **Stop**.

To complete the macro, follow the instructions in [Recording a Macro](#) on page 342 for replaying, identifying a logged out condition, and saving the macro.

Enhancing Macros

There are a number of optional enhancements that can be added to macros beyond the basic workflow.

[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](#).

[Insert loops](#)

Loops repeat selected portions of the macro until certain criteria is met or for a specified number of times. To insert a loop, select **Toolbox** → **Flow Control** → **For loop**. For more information, see [How to Insert and Modify Loops](#).

[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, select **Toolbox** → **Flow Control** → **If block**. To add an else condition, click the Add else link next to the If step title. For more details, see [TruClient Step Arguments](#).

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, select **Toolbox** → **Flow Control** → **Exit**.

[Insert comments](#)

You can insert comments into your macro by selecting **Toolbox** → **Misc** and dragging the Comment icon to the desired location.

[Insert 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. To insert a catch error step, select **Toolbox** → **Flow Control** → **Catch Error**.

Verify that an object exists

To verify that a string or object exists in the application, you can insert a verify step:

- 1 Select **Toolbox** → **Functions** and drag the Verify icon to the desired location.
- 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, select **Toolbox** → **Functions** → **Generic Object/Browser Action**, 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.

Debugging 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 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 **Breakpoints** .

Debug Macros Using Snapshots

You can use the snapshots generated during replay to debug macros by viewing the snapshots of the failed step(s).

- 1 Click **General Settings** .
- 2 Set the Replay Snapshot Generation setting to **On Error**.
- 3 Replay the macro.
- 4 Click **Snapshot View**  and in the Snapshot Viewer, click **Interactive Replay**. Note the step numbers of the steps that had errors.

You now have a group of snapshots in which errors occurred in the macro.

Modify and View Levels

Sometimes, steps that were recorded and are necessary for replay are placed in levels 2 and 3. In this case, you need to manually modify the level of those steps to level 1.

To modify a the macro's replay level, drag the slider in the toolbar to the desired level. Dragging the slider to level 3 displays and replays the steps on levels 1, 2, and 3.

To move a step to a different level, open the step and click on the step section. Move the slider to the desired level. If the step is part of a group step, both the group step and the individual step must be modified.

For more information, see Script Levels.

Insert Wait Steps

Wait steps cause the script to pause for a specified amount of time before continuing with the next step. Wait for Object steps cause the script 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, select **Toolbox** → **Functions** and drag the Wait or Wait for Object icon to the desired location in your script. 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.

Resolving Object Identification Issues

In dynamic Web sites, 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.

TruClient includes sophisticated mechanisms to overcome this challenge including the Highlight, Improve Object Identification, Replace Object, and Related Object options. When identifying objects for applications that recorded in windows, make sure that the correct window is selected using the Window tab. The following steps describe the ways to resolve these issues.

Highlight, Improve Identification, Replace, Related Objects all require the user to select an object in the application. There are cases in which various actions are required in the application to make the object visible such as mouse over and mouse click. In these cases 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 .

After you perform any of the changes, replay the single failed step in question and only afterwards replay the whole macro again. This will help verify whether the change has solved the issue you encountered.

The following paragraphs describe ways to resolve object identification issues.

Highlight an object

Regardless of which method of object identification is used, you can use the Highlight button



to check if an object is visible in the application at any time. If the object is not found, this may be an issue of pacing and timing. If the object cannot be found, an error message is displayed.

Object Identification

If the Highlight option fails, use **Improve Object Identification** . This will let TruClient 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.

Alternative Steps

Alternative steps allow you to view instances in which there are multiple ways to perform the same action in a step. If Improve Object Identification fails, try using one of the alternative steps.

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.

Before selecting one of the alternatives, try highlighting the object used by the alternative step and replaying it. This way you'll make sure the alternative step is replaying the necessary action.

Modify the Object Identification Method

You can modify the way TruClient identifies the object by modifying the object identification method in the Object section of the step properties. The following options are available:

- **Automatic.** TruClient's default object identification method. The Automatic method allows TruClient to use its internal advanced algorithms to locate the object. If this method does not successfully find the object during replay, click the Improve Object Identification button and replay the macro again.
- **XPath.** If Automatic identification fails, even after using Improve 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. Click the drop-down arrow next to the **XPath** edit box to select a suggested XPath for the object. You can manually modify the suggested path.

For example, if you need to select the first search result, regardless of the term being searched for, using XPath identification may help.

- **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 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 something like this:

```
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 [Debugging Macros](#) on page 350.

Relating objects to other objects

If the Improve Object Identification function does not solve the issue and neither do any of the alternative steps, 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. VuGen then uses this object to help locate the target object. To use this function, expand the step, select **Object → Related Objects**, and click



Add . 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 between identification methods.
- If several relations exist they all need to 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 made to the original step such as relations. Expand the step, select



Object, and click **Replace** . Select the new object and replay the macro.

Replace Object will tell TruClient that the object currently referenced in the step is incorrect. TruClient will remove any current knowledge of the object and learn the object you select. Therefore, you should only use the Replace Object option if the object you used during recording was the wrong one.

Inserting and Modifying 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 Functions 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. Loops arguments use JavaScript syntax. To insert a for loop, select **Toolbox** → **Functions** → **For Loop**.

"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 iteration in a for loop, the loop will end immediately without completing the remaining iterations. To insert a break statement, select **Toolbox** → **Functions** → **Break**.

"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, select **Toolbox** → **Functions** → **Continue**.

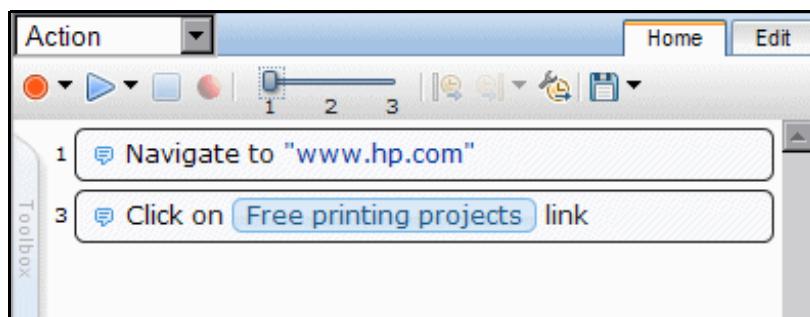
Script Levels

Some steps you perform while recording are not needed during replay. TruClient removes steps it deems to be unnecessary and places them in different script levels.

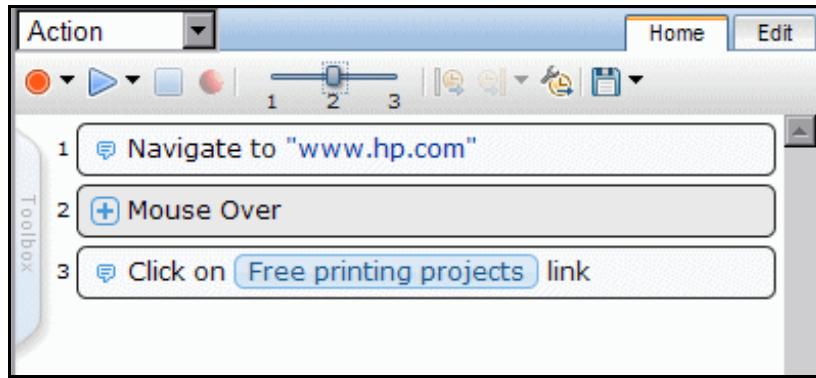
For example, a click step that occurs in an area of the application that has no effect is placed in level 2. During the replay phase, only steps that are visible are run. The default view displays level 1 steps only. To view steps from levels 2 and 3 as well, use the slide bar in the home tab.

In certain cases, you may want to manually change the level of a given step. This can happen in cases such as mouse-over steps (which are generally considered unnecessary and assigned to level 3).

The following illustration depicts a small script where the step numbers skip from 1 to 3. Step 2 is hidden in a different level.



After changing the display settings by using the slide bar, all steps are now displayed and will run if replayed in interactive mode.

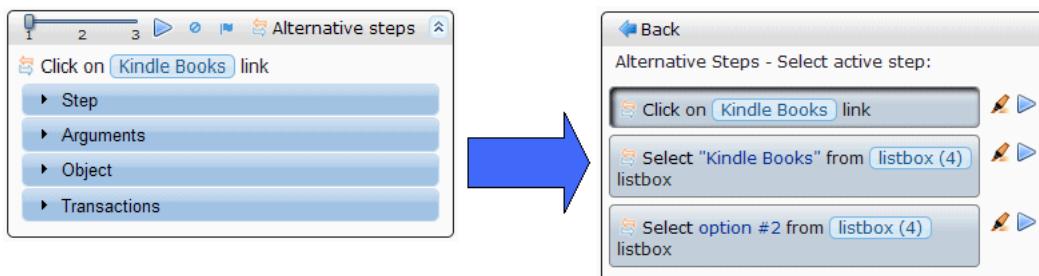


Alternative Steps

Alternative steps allow you to view instances in which there are multiple ways to perform the same action in a step. You can modify such steps to perform the given action to debug or enhance your macro.

Steps that have alternative options are labeled with an alternative step symbol . Click it to view the alternative options for that step. Click the desired alternative and select **Back**.

The illustration below depicts a step in which the second item in a drop-down list named “Kindle Books” was selected. The alternative steps feature gives you the option of defining the step based on clicking the link “Kindle Books,” selecting the object “Kindle Books” from the drop-down menu, or selecting the second item in the drop-down menu.



Snapshots

TruClient automatically generates snapshots during recording. These snapshots can be viewed by hovering the mouse over each step's icon. The snapshots are taken before the step's action is implemented. Click each snapshot to display it in a new browser tab. Make sure that the correct tab is active before replay.

You can also view snapshots by clicking **Snapshot View** .

Toolbox

The toolbox enables you to add steps to TruClient macros. The toolbox can be moved by dragging it up or down.

User interface elements are described in the following table

Toolbox User Interface Elements

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/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. <ul style="list-style-type: none">• 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 Enhancing Macros on page 349. 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. Evaluate C. Runs the C code contained in the step. Comment. A blank step that allows you to write comments in your macro.

Settings

Click **General Settings**  to open the *General Settings* dialog.

Proxy Settings

Proxy settings configured in the TruClient Web Macro Recorder are not used when TruClient is launched from the WebInspect Scan Wizard; TruClient will use whatever proxy settings are configured in the Scan Wizard.

Note: For Internet Explorer and Firefox, TruClient requests during a scan will not be sent to the proxy.

Select one of the following:

- **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 Browser Proxy** - Use the proxy options configured in your browser's Internet connection settings. Select your browser from the list.
- **Configure proxy using a PAC file** - Load proxy settings from a Proxy Automatic Configuration (PAC) file in the location you specify in the URL box.
- **Explicitly configure proxy** - Configure a proxy by entering the requested information.
 - **Type:** Select a protocol for handling TCP traffic through a proxy server. You can choose SOCKS4, SOCKS5, or standard.
 - **Server:** Enter the URL or IP address of your proxy server.
 - **Port:** Enter the port number (for example, 8080).
 - **Specify Alternative Proxy for HTTPS:** Select this option for proxy servers accepting HTTPS connections and then provide the requested server and port information.
 - **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.

Snapshot Generation

- Recording snapshots generation - Select **Never** or **Always**.
- Replay snapshots generation - Select **Never**, **On error**, or **Always**.

Replay Options

- **Maximum time for object-not-found** - Specify the maximum number of seconds that the macro recorder will wait for the target object of a replay step to appear.
- **Interstep interval** - Specify the minimum interval (in milliseconds) between steps.
- **End-of-network identification timeout** - Define the timeout (in milliseconds). The end-of-network for a step is recognized when the specified time has elapsed with no network activity.
- **Clean 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 Detection

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

The settings available for encryption are the same as those available in the standard Firefox browser. To view the Firefox documentation on encryption, see http://support.mozilla.com/en-US/kb/Options%20window%20-%20Advanced%20panel?as=u#w_encryption-tab.

[Encrypt Macro](#)

If you select this 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.

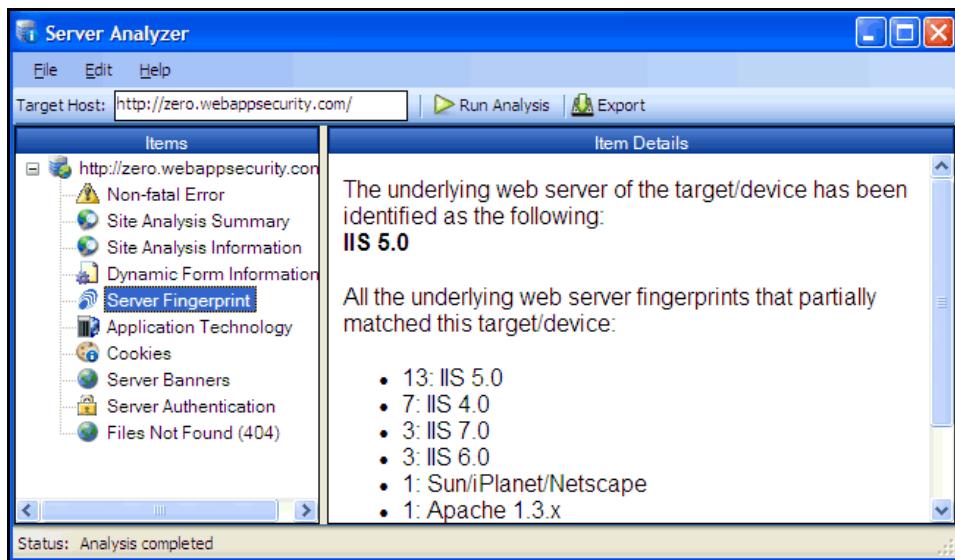
Server Analyzer

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

Analyzing a Server

Follow the steps below to analyze a server:

- 1 In the **Target Host** box, enter the URL or IP address of the target server.
- 2 If host authentication is required, or if you are accessing the host through a proxy server, select **Edit → Settings** and provide the requested information. See [Server Analyzer Settings](#) for detailed information.
- 3 Click the **Run Analysis** icon.



Server Analyzer Settings

Follow the steps below to modify the Server Analyzer settings:

- 1 Click **Edit → Settings**.
- 2 Select either the **Host Authentication** or **Proxy** category and enter the settings described in the following sections.
- 3 Click **OK**.

Authentication Method

If authentication is required, select a type from the **Authentication** list. See [Authentication Types](#) on page 177 for a description of the available authentication types.

Authentication Credentials

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

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

Proxy

Use these settings to access the Server Analyzer through a proxy server.

[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 Internet Explorer proxy settings](#)

Import your proxy server information from Internet Explorer.

[Use Firefox proxy settings](#)

Import your proxy server information from Firefox.

[Configure a proxy using a PAC file](#)

Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the **URL** box.

[Explicitly configure proxy](#)

Configure a proxy by entering 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. See Authentication Types on [page 177](#) for a description of the available authentication types.
- 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.

[HTTPS Proxy Settings](#)

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

Exporting Results

Follow the steps below to export the results of the analysis to an HTML file:

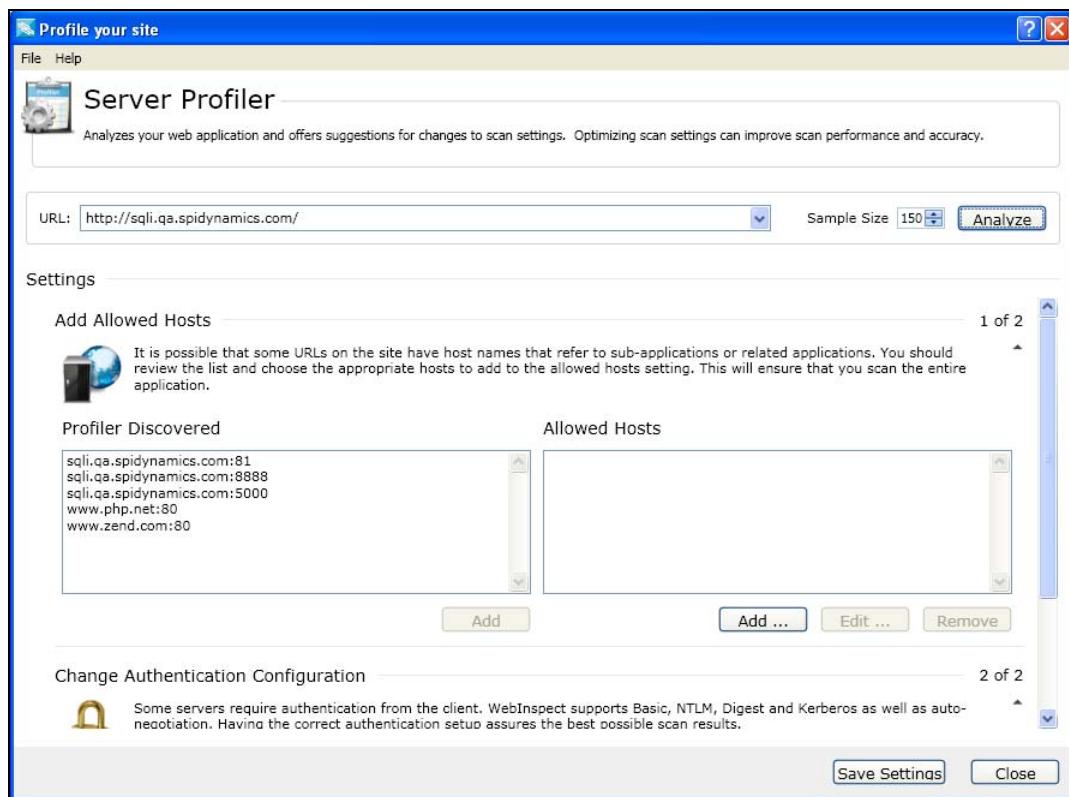
- 1 Click **File** → **Export**.
- 2 On the *Export File* window, select or enter a location and file name.
- 3 Click **Save**.

Server Profiler

Use the Server Profiler to conduct a preliminary examination of a Web site to determine if certain WebInspect settings should be modified. If changes appear to be required, the Server 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 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 Server Profiler determines that such a scheme has been implemented in the target site, it would suggest that you modify the WebInspect setting to accommodate this feature.



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

- Launch the Server Profiler as a standalone tool.
- Invoke the Server Profiler automatically when starting a scan.

Launching the Server Profiler as a Tool

Follow the steps below to launch the Server Profiler tool:

- 1 On the WebInspect menu bar, click **Tools** → **Server Profiler**.
- 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 Server 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.

Invoking the Server Profiler when Starting a Scan

Follow the steps below to launch the Server Profiler when beginning a scan:

1 Start a scan using one of the following methods:

- On the WebInspect **Start** page, click **Start a Web Site Scan**.
- Click **File** → **New** → **Web Site Scan**.
- Click the drop-down arrow on the **New** icon (on the toolbar) and select **Web Site Scan**.
- On the WebInspect **Start** page, click **Manage Scheduled Scans**, click **Add**, and then select **Web Site 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**.

SWFScan

The HP Web Security Research Group developed SWFScan to 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.

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.

Vulnerability Detection

HP SWFScan tests for the following Flash security vulnerabilities. Checks for additional vulnerabilities will be added to SecureBase (through Smart Update) as they are developed.

ActionScript 3 Vulnerabilities Detected by SWFScan

HP SWFScan finds the following vulnerabilities in applications built on Flash 9 and above.

Insecure Programming Practice

- Insecure Security.allowInsecureDomain() usage
- Insecure Security.allowDomain() usage
- Insecure LocalConnection.allowDomain() usage
- Insecure Flash Storage Object usage
- Shared Flash Storage Object usage
- Possible Malicious Activity (LoadBytes)
- Interesting Package/Class/Function Names
 - Potential User Account Information
 - Possible Commerce Information
 - Possible Cryptographic Data
 - Potential Personal Information
 - Possible Application Information
 - Potentially Interesting Name Encountered

Insecure Application Deployment

- Complete Flash Application Source Available
- Debugging Information (trace function)
- Debugging Information (Source file disclosure)
- Remote Flash Debugging Enabled

Adobe Best Practices Violation

- Minimum Stage Size For Security Dialogs
- Utilize AllowScript Privileges
- Utilize AllowNetworking Privileges
- Utilize AllowFullscreen Privileges

Information Disclosure

- Possible Social Security Number
- Possible Credit Card Number Disclosure
- Internal IP Disclosure
- Path Disclosure (win32)
- Path Disclosure (unix)
- PGP Public Key Block Detected
- PGP Private Key Block Detected
- MD5 Hash Detected
- SHA-0/SHA-1 Hash Detected
- SQL Query Detected
- LDAP Query Detected
- XPath Query Detected
- Data Connection String
 - Generic
 - MSSQL ODBC Trusted Connection
 - MSSQL OleDb Trusted Connection
 - MSSQL via IP Address
 - MSSQL .NET DataProvider Standard Connection or Sybase .NET DataProvider
 - MSSQL .NET DataProvider Trusted Connection
 - MSSQL .NET DataProvider via IP Address
 - Access and Oracle ODBC -- Standard Security for MS Access and ODBC Oracle Driver
 - Access ODBC Workgroup - System Database
 - Access OleDb with MS Jet Workgroup - System Database
 - Access OleDb with MS Jet With Password
 - Oracle ODBC New Microsoft Driver

- Oracle ODBC Old Microsoft Driver
- Oracle OleDb Microsoft Driver and Oracle Driver - possible trusted connection
- Oracle OleDb Oracle Driver - Trusted Connection
- Oracle .NET DataProvider from Microsoft and Oracle - Standard Connection
- Oracle .NET DataProvider from Microsoft and Oracle - Trusted Connection
- IBM DB2 ODBC without DSN and OleDb IBM Driver
- IBM DB2 OleDb Microsoft Driver
- IBM DB2 .NET DataProvider from IBM
- MySQL ODBC MyODBC Driver - local database
- MySQL ODBC MyODBC Driver - remote database
- MySQL .NET DataProvider from CoreLab
- Sybase ODBC Sybase System 12 (12.5) ODBC Driver
- Sybase ODBC Sybase System 11 ODBC Driver or Intersolv 3.10 ODBC Driver
- Sybase ODBC SQL Anywhere
- Sybase OleDb Sybase Adaptive Server Enterprise (ASE)
- Informix ODBC DSN INFORMIX 3.30 ODBC Driver
- Informix ODBC without DSN INFORMIX 3.30 ODBC Driver
- Informix OleDb IBM Informix OleDb Provider

ActionScript 1 and 2 Vulnerabilities Detected by SWFScan

HP SWFScan finds the following vulnerabilities in applications built on Flash 8 and below.

Possible Cross-Site Scripting

- Identifying undefined global variables
- Identifying injection points for cross-site scripting vectors
 - getUrl
 - XML.load
 - loadMovie/loadMovieNum
 - htmlText
 - ExternalInterface

Dangerous functions accepting user supplied data

- loadVariables/LoadVars
- System.Security.loadPolicyFiles
- Insecure Programming Practice
- Insecure Security.allowInsecureDomain() usage
- Insecure Security.allowDomain() usage
- Insecure LocalConnection.allowDomain() usage

- Insecure Flash Storage Object usage
- Shared Flash Storage Object usage
- Possible Malicious Activity (LoadBytes)
- Interesting Package/Class/Function Names
 - Potential User Account Information
 - Possible Commerce Information
 - Possible Cryptographic Data
 - Potential Personal Information
 - Possible Application Information
 - Potentially Interesting Name Encountered

[Insecure Application Deployment](#)

- Debugging Information (trace function)
- Debugging Information (Source file disclosure)
- Remote Flash Debugging Enabled

[Information Disclosure](#)

- Possible Social Security Number
- Possible Credit Card Number Disclosure
- Internal IP Disclosure
- Path Disclosure (win32)
- Path Disclosure (unix)
- PGP Public Key Block Detected
- PGP Private Key Block Detected
- MD5 Hash Detected
- SHA-0/SHA-1 Hash Detected
- SQL Query Detected
- LDAP Query Detected
- XPath Query Detected
- Data Connection String
 - Generic
 - MSSQL ODBC Trusted Connection
 - MSSQL OleDb Trusted Connection
 - MSSQL via IP Address
 - MSSQL .NET DataProvider Standard Connection or Sybase .NET DataProvider
 - MSSQL .NET DataProvider Trusted Connection
 - MSSQL .NET DataProvider via IP Address
 - Access and Oracle ODBC -- Standard Security for MS Access and ODBC Oracle Driver

- Access ODBC Workgroup - System Database
- Access OleDb with MS Jet Workgroup - System Database
- Access OleDb with MS Jet With Password
- Oracle ODBC New Microsoft Driver
- Oracle ODBC Old Microsoft Driver
- Oracle OleDb Microsoft Driver and Oracle Driver - possible trusted connection
- Oracle OleDb Oracle Driver - Trusted Connection
- Oracle .NET DataProvider from Microsoft and Oracle - Standard Connection
- Oracle .NET DataProvider from Microsoft and Oracle - Trusted Connection
- IBM DB2 ODBC without DSN and OleDb IBM Driver
- IBM DB2 OleDb Microsoft Driver
- IBM DB2 .NET DataProvider from IBM
- MySQL ODBC MyODBC Driver - local database
- MySQL ODBC MyODBC Driver - remote database
- MySQL .NET DataProvider from CoreLab
- Sybase ODBC Sybase System 12 (12.5) ODBC Driver
- Sybase ODBC Sybase System 11 ODBC Driver or Intersolv 3.10 ODBC Driver
- Sybase ODBC SQL Anywhere
- Sybase OleDb Sybase Adaptive Server Enterprise (ASE)
- Informix ODBC DSN INFORMIX 3.30 ODBC Driver
- Informix ODBC without DSN INFORMIX 3.30 ODBC Driver
- Informix OleDb IBM Informix OleDb Provider

Analyzing Flash Files

You can use SWFScan as a standalone tool or as an integrated component of WebInspect.

Analyze a Flash file using SWFScan as a standalone tool

- 1 Launch SWFScan:

Click **Start → All Programs → HP → HP Security Toolkit → SwfScan**.

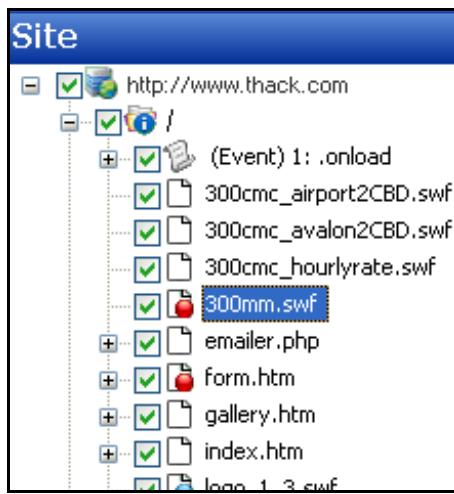
- 2 Specify the Flash file (.swf) you want to analyze.

- In the **Path or URL** combo box, enter or select the full path to a Flash file and click  on the SWFScan toolbar.
- or -
- Click **File → Open**, select a Flash file from a local storage device, and click **Open**.
SWFScan loads and decompiles the selected file.

- 3 Click  on the SWFScan toolbar.

Analyze a Flash file using SWFScan as an integrated component of 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.

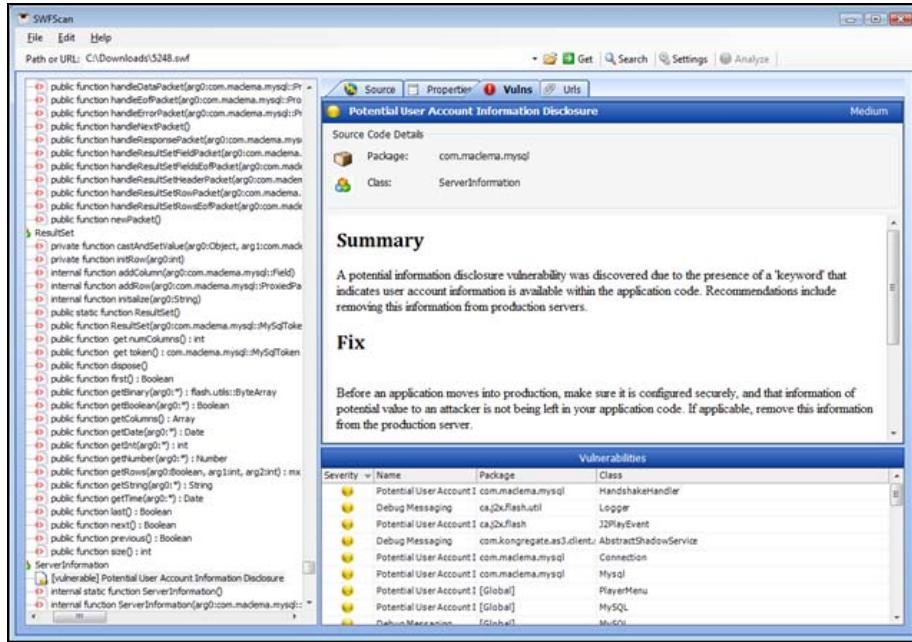
- 2 Click  on the SWFScan toolbar.

➤ 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).

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.



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** box, 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.

SWFScan Settings

Use the following procedure to configure SWFScan settings.

- 1 Click  on the SWFScan toolbar.
- 2 Click any of the four tabs presented, which are described below.

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:

- a Click **Add**.
- b On the *Add Exclusion Rule* window, enter a name for the rule and a regular expression that describes the package.
- c 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:

- a Click **Add**.
- b On the *Add Exclusion Rule* window, enter a name for the rule and a regular expression that describes the package or class.
- c 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, SWFScan uses the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and configure the browser's Web proxy settings.
- **Use Internet Explorer proxy settings**—Select this option to import your proxy server information from Internet Explorer.
- **Use Firefox proxy settings**—Select this option to import your proxy server information from Firefox.

- **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 box.
- **Explicitly configure proxy**—Select this option to configure a proxy manually, and then enter the requested information.
- **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. You can enable or disable a check by selecting or clearing its associated check box.

- 3 When complete, click **OK**.

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

Report Designer

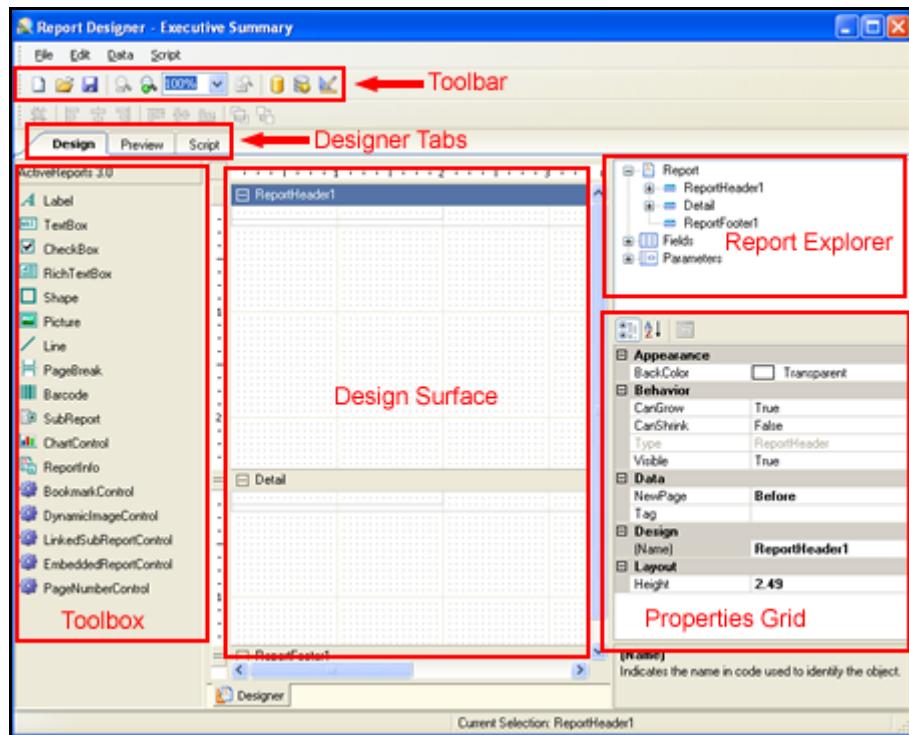
The Report Designer is an HP integration of the ActiveReports® 3.0 report designer developed by Grape City - Data Dynamics. It provides the ability to create and modify reports.

For complete information on using ActiveReports, refer to the ActiveReports User Guide and Class Library.

User Interface

The Report Designer contains six main components, as depicted in the following illustration:

- Toolbar
- Designer Tabs
- Toolbox
- Design Surface
- Report Explorer
- Properties Grid



Toolbar

The Report Designer toolbar is illustrated below.



Report Designer Toolbar

Icon	Function	Description
	New	Opens the <i>Create Report Definition</i> window, allowing you to select the queries to be included in the report.
	Open	Opens the <i>Open a Report</i> window, allowing you to select a report or subreport for editing.
	Save	Saves the open report.
	Zoom In	Increases the magnification of the design surface at 50 percent increments.
	Zoom Out	Decreases the magnification of the design surface at 50 percent increments.
	Magnification Percentage	Allows you to select a magnification setting for the design surface.
	Actual Size	Returns the magnification of the design surface to 100 percent.
	Set Data Source	Allows you to specify the scan that will provide the data.
	Set Custom Data Source	Allows you to specify a custom data source.
	Parameter Designer	Opens the Parameter Designer tool.

Menus

The Report Designer contains the following menus:

Report Designer Menus

Menu	Command	Description
File	New	Opens the <i>Create Report Definition</i> window, allowing you to select a definition for a new report.
	Open	Opens the <i>Open a Report</i> dialog, allowing you to select a report for editing.
	Save	Saves the open report.
	Save As	Saves the open report to a file you specify.
	Export	Saves the report in a format you specify.
	Enable Console Output	If enabled, WebInspect presents a pane (at the bottom of the window) that displays the status of each report page being generated. If a problem is encountered, this pane displays an exception message and stack trace. This pane is also visible on the Preview tab of the Report Designer.
	Exit	Terminates the Report Designer.
Edit	Parameter Designer	Opens the Parameter Designer tool.
	Modify/Create Report	Opens the <i>Modify Report Definition</i> dialog, allowing you to change the report definition.
	Delete	Deletes the selected object.
	Cut	Deletes the selected object and saves it to the clipboard.
	Copy	Copies the selected object to the clipboard.
	Paste	Inserts the contents of the clipboard.
	Undo	Reverses the last operation performed.
Data	Redo	Reverses the last Undo operation.
	Set Scan and Report Inputs	Allows you to select a scan and specify report parameters.
	Set Custom Data Source	Opens the <i>Report Data Source</i> dialog, allowing you to connect to various sources.
	Edit Global Styles	Opens the Report Styles Editor. Use this to create or modify a stylesheet.
	Edit Report Styles	Opens the Report Styles Editor. Use this to create or modify styles for the report on which you are currently working
	Edit Report Settings	Opens the <i>Report Settings</i> dialog, allowing you to modify many facets of your report.
	Import	Allows you to select a script from the script library to import into the designer.

Report Designer Menus (cont'd)

Menu	Command	Description
	Compile	Compiles the script.
	Find	Opens the Script tab and presents the <i>Find/Replace</i> dialog, allowing you to search for the text you specify.
	Script Editor	Opens the Script Editor.

Designer Tabs

The Report Designer contains the following three tabs.

Design Tab

By default, when you create or open a report, the Design tab is selected. Use this area to perform all design-time and run-time functions associated with your report, such as creating a layout, binding to data sources, creating event-handling methods, and more.

Script Tab

Selecting the Script tab opens the script editor, which gives you the ability to add scripting to your report. The Script editor allows you to create event-handling methods. In the Report Events tab on the right, there is a combo box where you can select any report section to attach an event-handling method.

Preview Tab

The Preview tab allows you to view what your report looks like at run time with actual scan data. This makes it easy to quickly see the run-time impact of changes you make in the designer or the code-behind. Use the Preview toolbar to navigate the report and add annotations.

Toolbox

The toolbox displays a variety of controls. To add a control, drag it from the toolbox and drop it on the design surface (canvas), where you can modify its size, position, alignment, and properties.

- Barcode — Inserts an ActiveReports Barcode control; can be bound to a database field.
- ChartControl — Inserts a chart in any of a variety of styles.
- Checkbox — Inserts a check box; can be bound to a database field.
- Label — Inserts a new static label control; can be bound to a database field.
- Line — Inserts a line control.
- PageBreak — Inserts a page break within a selection.
- Picture — Inserts an image loaded from a file; can be bound to a database field.
- ReportInfo — Displays report information in a number of format strings such as {PageNumber} or {PageCount}: can be bound to a database field.
- Textbox — Inserts a textbox; can be bound to a database field
- Shape — Inserts a rectangle, circle or square shape.

- Subreport — Inserts a Subreport control to link to another report.
- RichTextBox — Inserts an ActiveReports RichTextBox control; can be bound to a database field.
- BookmarkControl — Inserts a hyperlink in the table of contents; clicking the hyperlink navigates to the bookmark.

Note: Bookmark text can be formatted as follows:

{=MainReportName}\<static-text>\{=<field-name>}

where

MainReportName is optional (and doesn't need to appear first)

\ indicates the beginning of a hierarchical level

<static-text> is any text you assign to the bookmark

<field-name> is the name of a bound or calculated field

- DynamicImageControl — Allows you to associate an image selector control with an image (using the Parameter Designer), so the user can select an image at run time. Can be bound to a database field.
- LinkedSubreportControl — Creates a link to the subreport you select. Use the AssociatedFields property to pass values to the subreport.
- EmbeddedReportControl — Allows you to design a subreport “on the fly” (rather than using a LinkedSubreportControl) using the DataTableField property.
- PageNumberControl — Allows you to place a page number in the report (usually in the page footer).

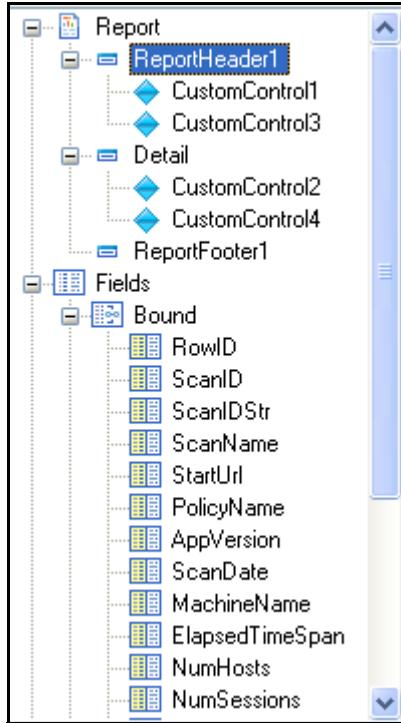
Design Surface

The default design surface contains the following base components:

- PageHeader section--This section can be used to print column headers, page numbers, page titles, or any information that needs to be printed once at the top of each page. Bound controls in the PageHeader or PageFooter are not supported. The data in such controls may not be in sync with the data displayed in other sections on the page.
- Detail section--This section is the body of the report that prints once for each record in the data source. A report's layout may contain only one Detail section.
- PageFooter section--This section can be used to print page totals, page numbers or any other information that needs to be printed once at the bottom of each page.
- Designer/Script/Preview tabs--The Designer and Script tabs can be clicked to toggle between design and script views, while the Preview tab allows for a fully functional design-time preview of how a report will look and behave at run time.

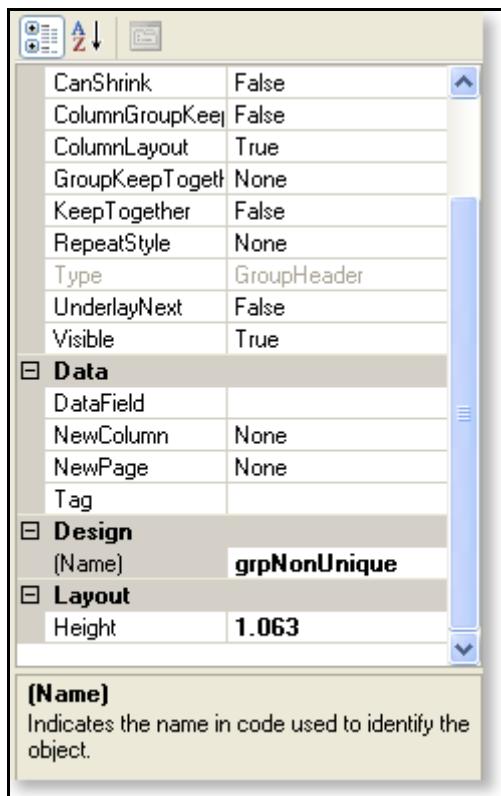
Report Explorer

The Report Explorer serves as the information focal point for your report. From it, you can gain a quick overview of the elements that compose the report, remove individual controls, add parameters and calculated fields, bind data fields to text box controls, and modify properties and report behavior via the Properties grid.



Properties Grid

The Properties Grid allows you to view or modify properties for an object selected on either the Design Surface or the Report Explorer.



Creating a Report

- 1 Open or create a report definition.

To create a report definition:

- a Click **File** → **New** (or click the New icon on the toolbar).
- b Create a report definition.
- c Enter a name and (optionally) a brief description for the report.
- d Select a report context: either **Scan** or **Session**.

When a scan is open, users can generate a session report by right-clicking a session and selecting **Generate Session Report** from the context menu.

- e If you want the report name to be included in the list of WebInspect reports, select **Exposed in Product**.

Typically, you do not select this option if you are creating a subreport.

- f If you are creating a header/footer template, select **Header/Footer Template**.

- g Select one or more views from the View Name list. To see the view parameters and fields, click the view name.

- h Click **OK**.

To open a report definition:

- a Click **File** → **Open** (or click the Open icon on the toolbar).

- b Select a report or subreport.

- c Click **OK**.

- 2 Design your report. For complete information on using ActiveReports, refer to the ActiveReports User Guide and Class Library.

- 3 To modify the script associated with this report, click the **Script** tab.

- 4 To modify or create parameters associated with this report, click **Edit** → **Parameter Designer**.

- 5 To modify the styles associated with this report, click **Data** → **Edit Report Styles**.

- 6 To preview your work:

- a Click the **Preview** tab.

- b On the *Generate a Report* dialog, select a scan and click **Next**.

- c If the report includes parameters, select parameters.

- d Click **Finish**.

Report Script Editor

Use the Report Script Editor to create or modify scripts maintained in a script library. You can then import these scripts into reports.

All scripts must be written using the C# language.

The Report Script Editor menu bar contains the following menus:

Report Script Editor Menus

Menu	Command	Description
File	Save	Save the script to a library.
	Refresh	Redisplay the script.
	Exit	Terminate the Script Editor.
Edit	Find	Open a <i>Find/Replace</i> dialog, allowing you to search for and optionally replace text in the script.
Script	Import	Incorporate a script library into the script you are developing.
	Compile	Compile the script.
Help	Help	Open the Help file.

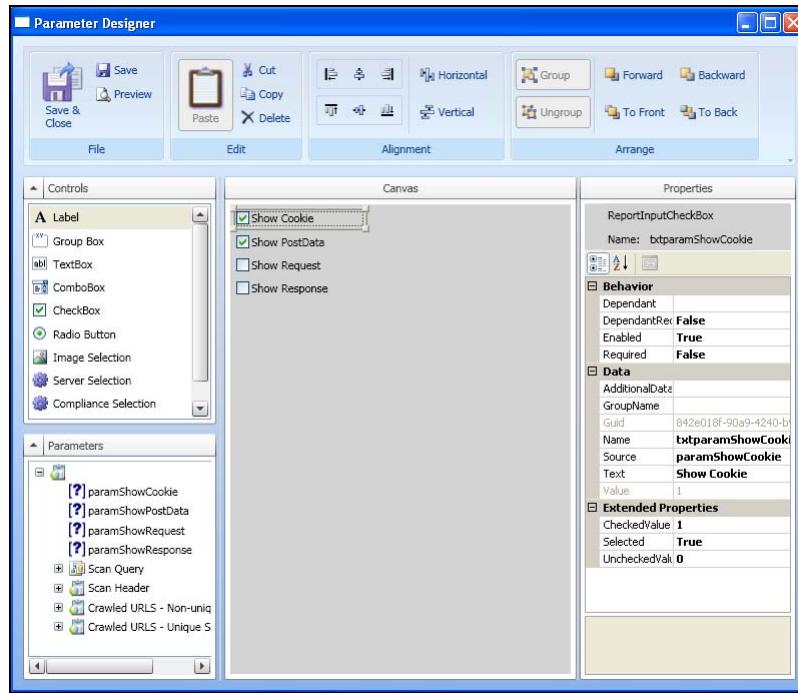
Parameter Designer

Reports have three types of inputs that can be used for filtering data or supplying custom content to reports. They are:

- **Data View parameters (query parameters)**—Data View parameters are used to pass values to the underlying Data View of the report for filtering data. Parameter names begin with @.
- **Report Parameters**—Report parameters are used to pass values entered by the user to the report. These values are then used by the report to alter report behavior or format.
- **Replacements**—Replacements are tokens that exist in the data view. Replacement inputs are used to pass values to these tokens. Replacements are used to change the sort order of a data view or to provide additional criteria to the data view.

Users have the opportunity to provide values for these inputs when generating a report. Before a user can be prompted to enter inputs, however, report designers must specify which inputs will be displayed to the user and how they will be presented. This is accomplished by using the Parameter Designer.

To open the Parameter Designer, from an open report in the Report Designer, click the Parameter Designer icon  on the toolbar or choose **Parameter Designer** from the **Edit** menu.



The Parameter Designer has five areas.

Toolbar

The toolbar provides easy access to all of the functions of the designer:

- **Save and Close**—Saves the current design to the report and closes the parameter designer window.
- **Save**—Saves the current design to the report.
- **Preview**—Opens a window showing what the designed inputs will look like at run time.
- **Cut, Copy, Paste, Delete**—Manipulate controls on the canvas.
- **Alignment**—Align one or more selected controls on the canvas.
- **Group/Ungroup**—A designer can group two or more selected controls on the canvas. When controls are grouped together, they can be moved together on the canvas.
- **Forward**—Bring the selected control forward one layer.
- **Backward**—Send the selected control backward one layer.
- **To Front**—Bring the selected control to the top most layer.
- **To Back**—Send the selected control to the bottom most layer.

Canvas

The canvas is the design area, which constitutes a visual representation of the parameters that are presented at run-time. Controls can be added, modified, and deleted from the canvas.

Properties Grid Pane

This area displays the properties of object(s) selected in the design canvas or the Parameters pane, whichever has the focus.

Controls Toolbox

The Controls toolbox lists the types of controls that may be added to the report. They include, in addition to the standard self-explanatory controls, the following special controls:

- **Server Selection**—A drop-down list of available servers in the selected scan.
- **Compliance Selection**—A list of compliance templates; suitable for compliance reports only.
- **Sort Control**—Allows you to select how you want the report data to be sorted.

To add a control, drag it from the toolbox and drop it on the canvas.

Report Parameters Pane

This pane displays a hierarchical representation of all parameters available to the current report and its subreports. Icons indicate the parameter type.

Query 

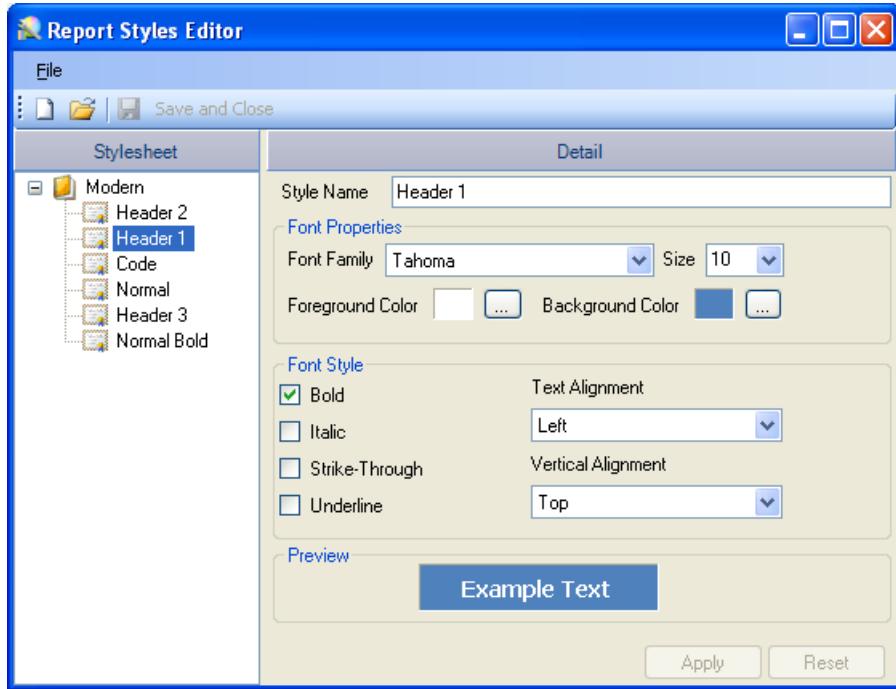
Report 

Replacement 

Report Styles Editor

When creating or modifying a report, the Report Designer uses the stylesheet that is specified as the default. If you want to create or modify styles for the report on which you are currently working, select **Edit Report Styles** from the **Data** menu. New styles will be added to the report; modified styles will override the default definition for this report.

Conversely, if you want to create or modify a stylesheet, select **Edit Global Styles** from the **Data** menu. You can then edit or create stylesheets, and specify the stylesheet that will be initially assigned to all reports as the default.



Report Structure

Report Structure

A report section contains a group of controls that are processed and printed at the same time as a single unit. ActiveReports defines the following section types.

Report Header

A report can have one report header section that prints at the beginning of the report. This section generally is used to print a report title, a summary table, a chart or any information that needs only to appear once at the report's start.

Report Footer

A report can have one report footer section that prints at the end of the report. This section is used to print a summary of the report, grand totals, or any information that needs to print once at the report's end.

Page Header

A report can have one page header section that prints at the top of each page. Unless the page contains a report header section, the page header will be the first section that prints on the page. The page header section is used to print column headers, page numbers, a page title, or any information that needs to appear at the top of each page in the report.

Page Footer

A report can have one page footer section that prints at the bottom of each page. It is used to print page totals, page numbers, or any other information that needs to appear at the bottom of each page.

Group Header/Footer

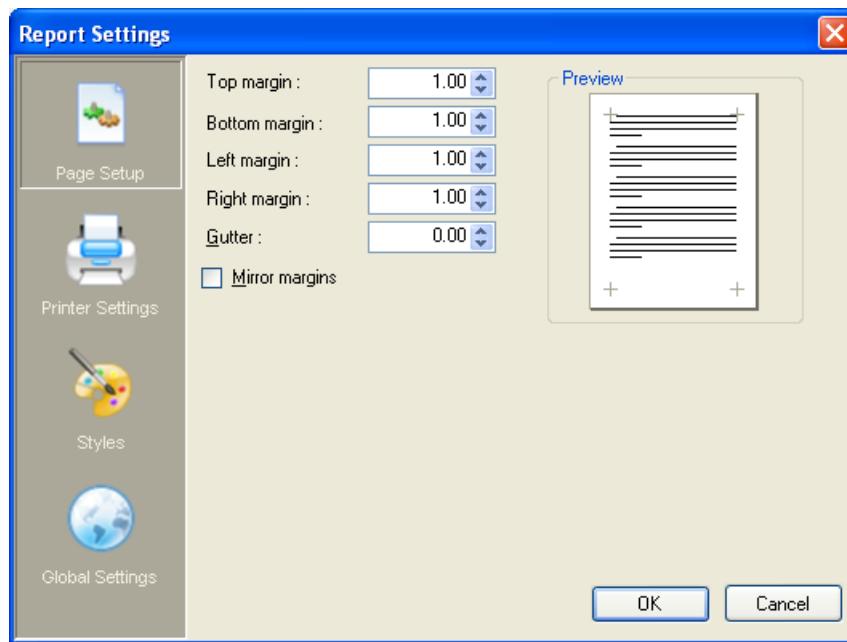
A report can consist of single or multiple nested groups, with each group having its own header and footer sections. The header section is inserted and printed immediately before the detail section. The footer section is inserted and printed immediately after the detail section.

Detail

A report has one detail section. The detail section is, in some cases, the body of the report and one instance of the section is created for each record in the report.

Report Settings

You can modify facets of your report, such as the page setup, printer settings, styles, and global settings of your report at design time. To make changes, access the *Report Settings* dialog by selecting Data > Edit Report Settings.



Charts

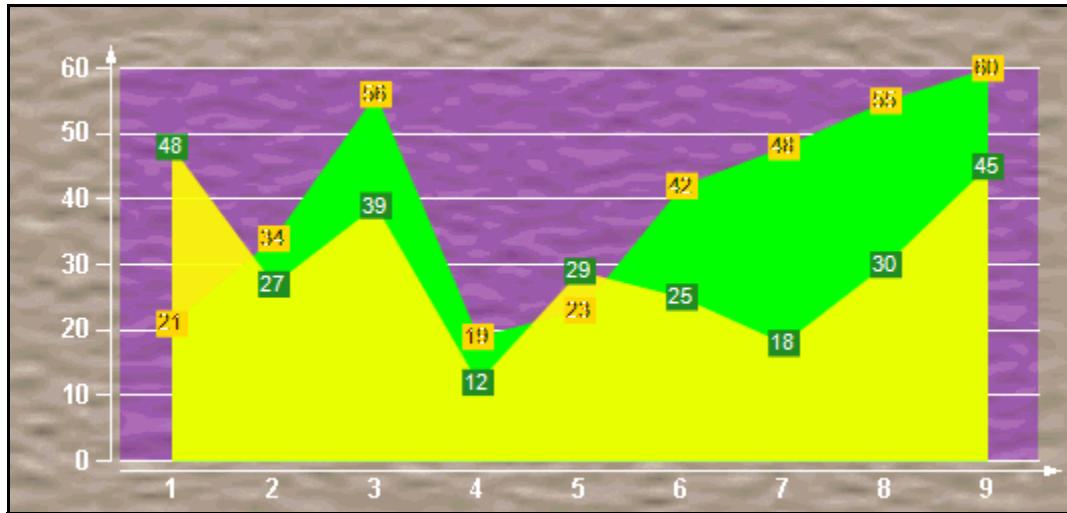
Chart Types

Chart types include Common Charts, 3D Charts, and XY Charts. See the on-line Help for more extensive illustrations of chart types.

Common Charts

- **Area Charts**

Use an area chart to compare trends over a period of time or in specific categories.



Number of Y values/data points: 1

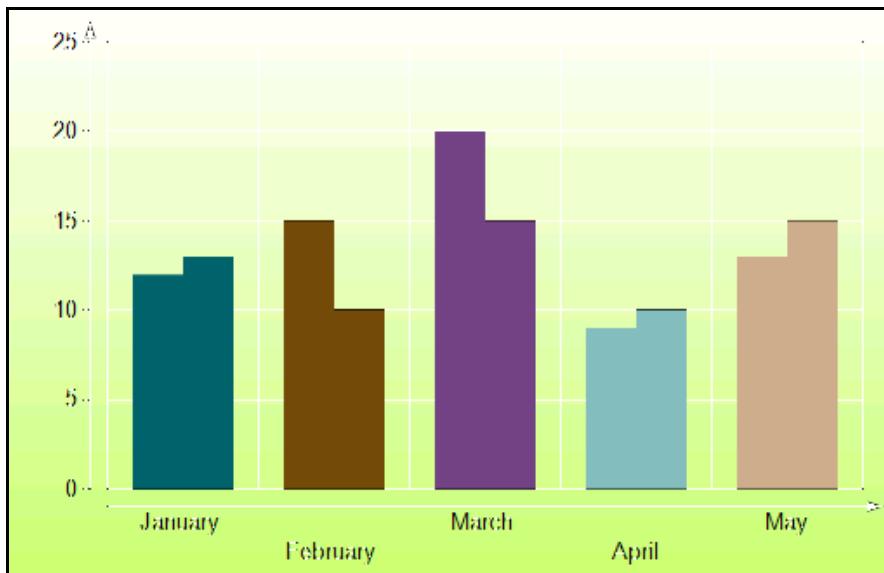
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **Bar2D Charts**

Use a bar chart to compare values of items across categories.



Number of Y values/data point: 1

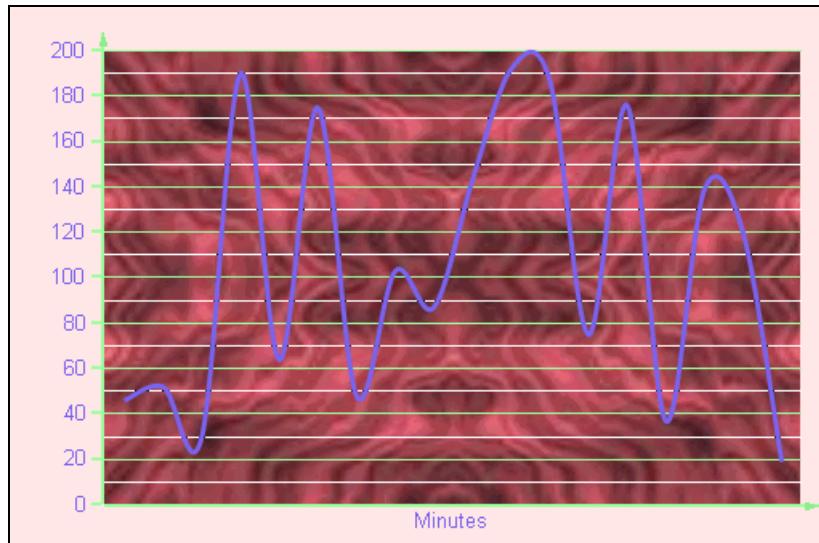
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value

- **Bezier Charts**

Use a Bezier or spline chart to compare trends over a period of time or in certain categories. It is a line chart that plots curves through the data points in a series.



Number of Y values/data point: 1

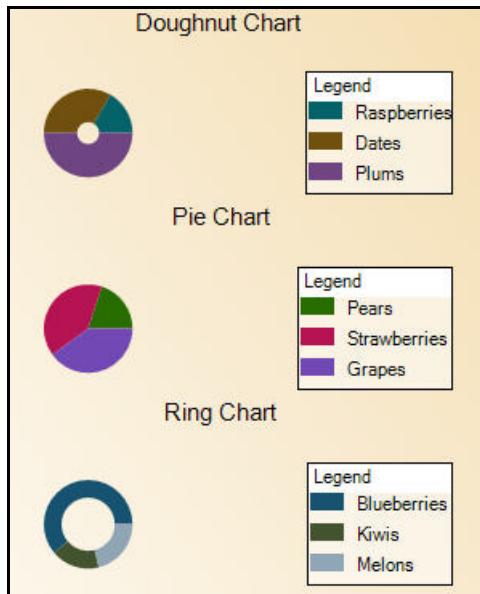
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **Doughnut/Pie Charts**

A doughnut chart shows how the percentage of each data item contributes to the total.



Number of Y values/data point: 1

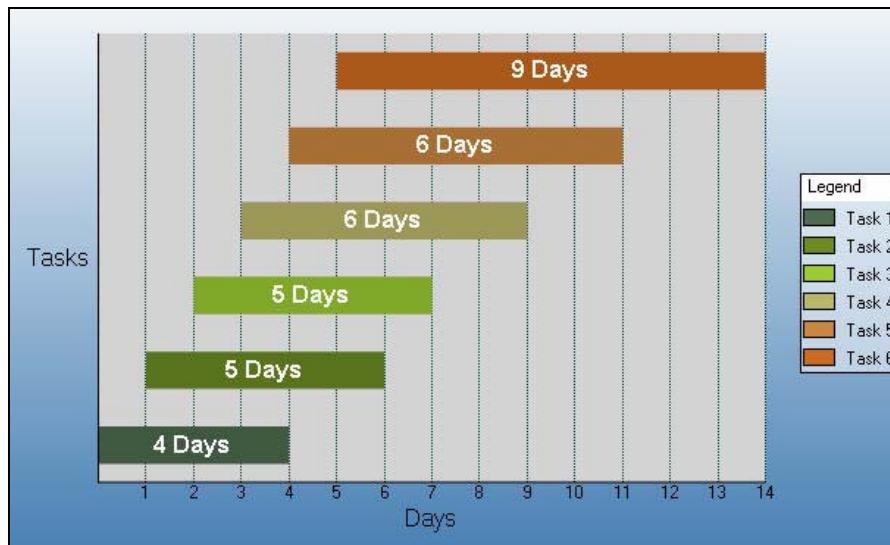
Number of Series: 1

Marker Support: Series or Data Point

Custom Properties: ExplodeFactor gets or sets the amount of separation between data point values. HoleSize gets or sets the inner radius of the chart. OutsideLabels gets or sets a value indicating whether the data point labels appear outside the chart. StartAngle gets or sets the horizontal start angle for the series.

- **Gantt Charts**

The Gantt chart is a project management tool used to chart the progress of individual project tasks. The chart compares project task completion to the task schedule.



Number of Y values/data point: 2

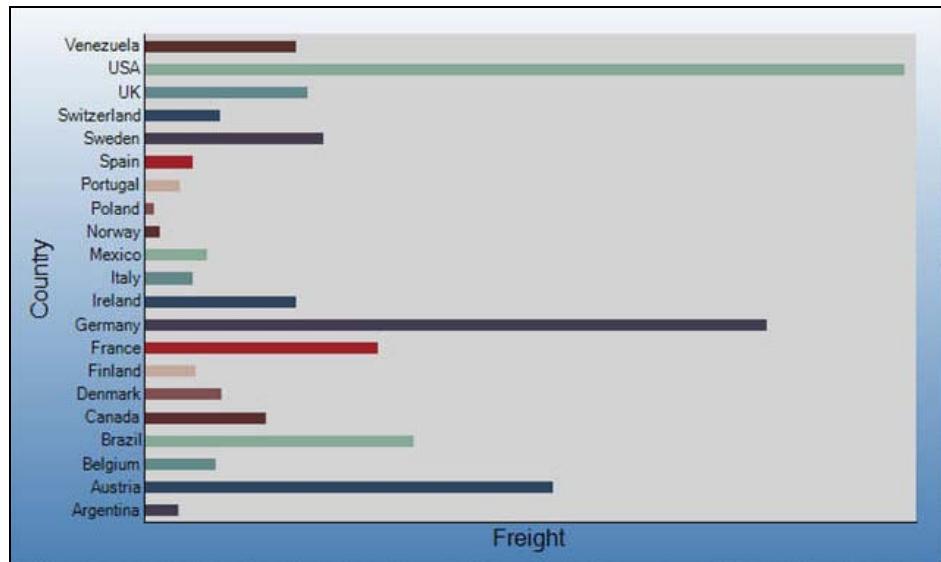
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value.

- **HorizontalBar Charts**

Use a horizontal bar chart to compare values of items across categories with the axes reversed.



Number of Y values/data point: 1

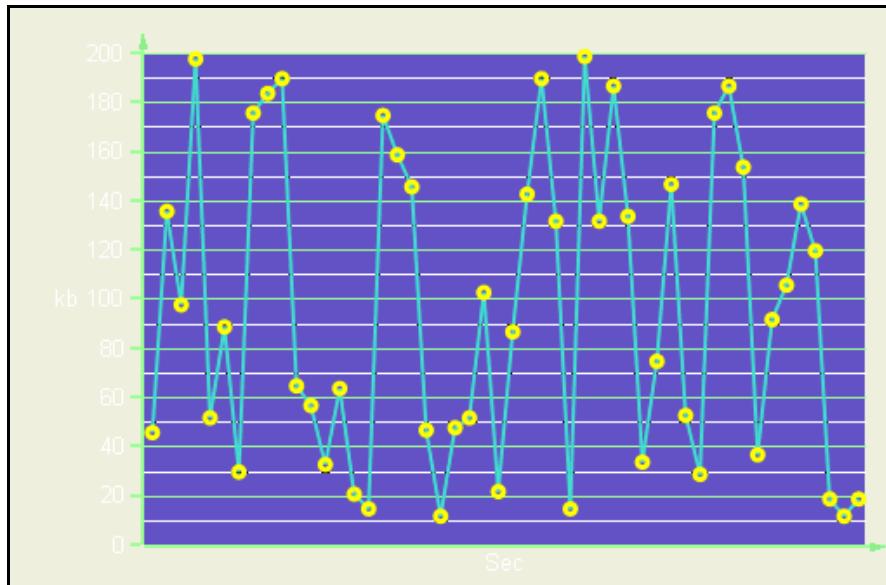
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value.

- **Line Charts**

Use a line chart to compare trends over a period of time or in certain categories.



Number of Y values/data point: 1

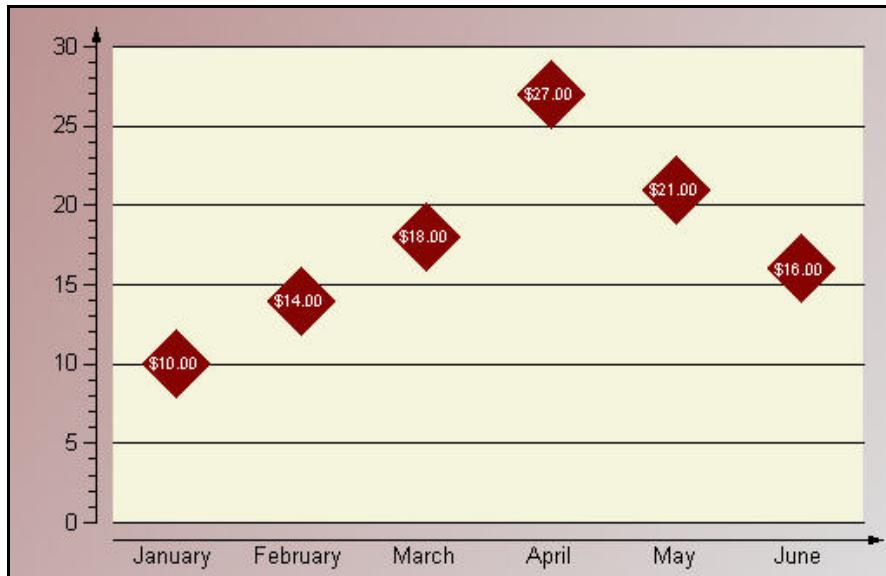
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **Scatter Charts**

Use a scatter chart to compare values across certain categories.



Number of Y values/data point: 1

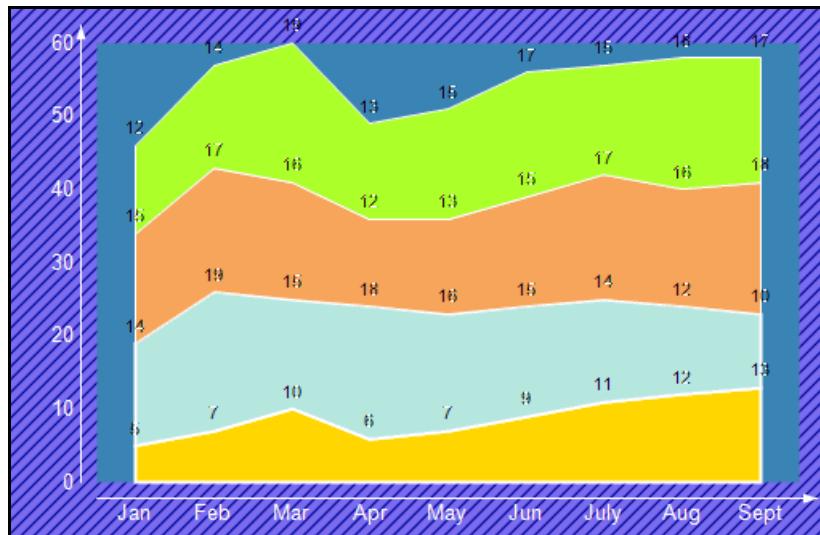
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **StackedArea Charts**

A stacked area chart is an area chart with two or more data series stacked one on top of the other. Use this chart to show how each value contributes to a total.



Number of Y values/data point: 1

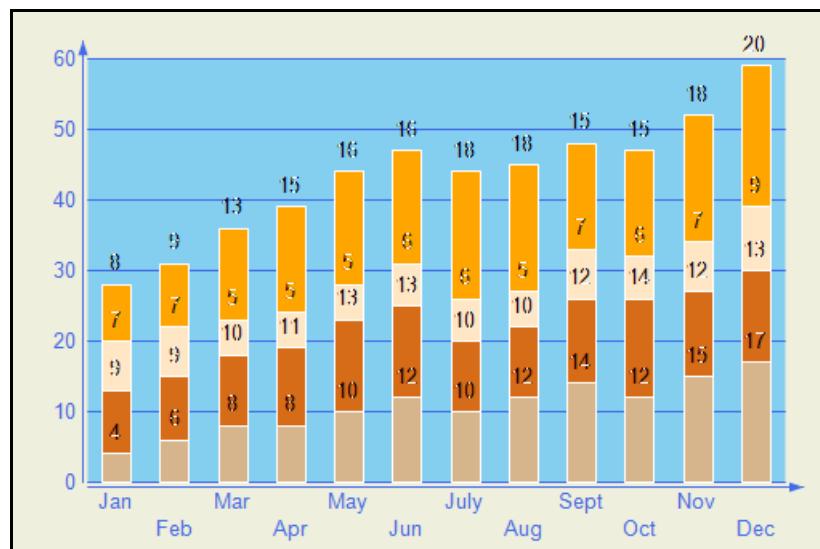
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **StackedBar Charts**

A stacked bar chart is a bar chart with two or more data series stacked one on top of the other. Use this chart to show how each value contributes to a total.



Number of Y values/data point: 1

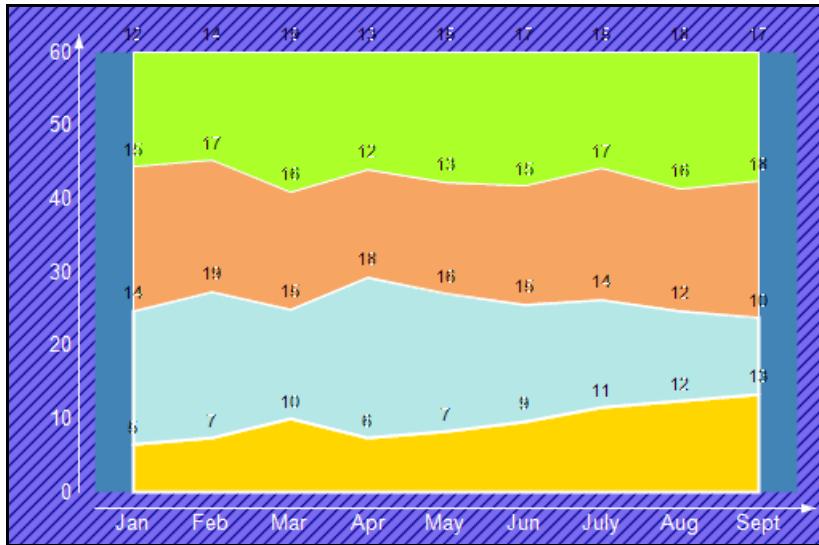
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value

- **StackedArea100Pct Charts**

A stacked area 100 percent chart is an area chart with two or more data series stacked one on top of the other to sum up to 100 percent. Use this chart to show how each value contributes to a total with the relative size of each series representing its contribution to the total.



Number of Y values/data point: 1

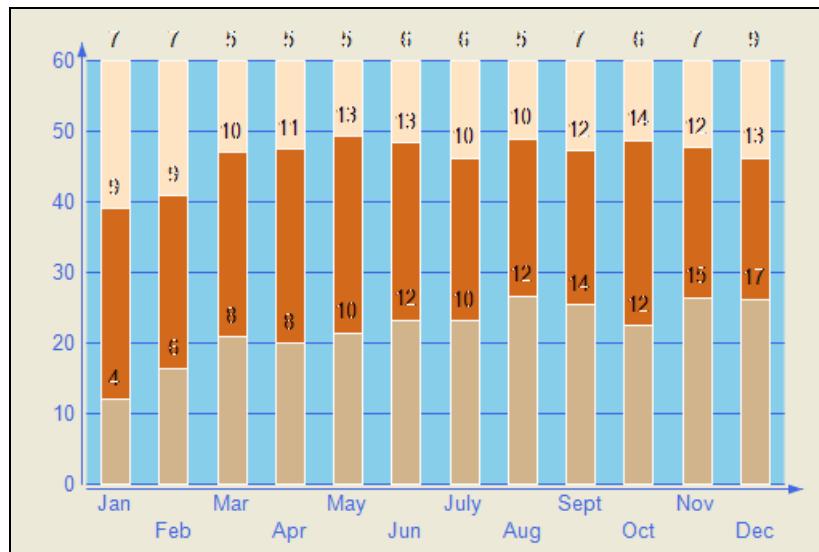
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **StackedBar100Pct Charts**

A StackedBar100Pct chart is a bar chart with two or more data series stacked one on top of the other to sum up to 100 percent. Use this chart to show how each value contributes to a total with the relative size of each series representing its contribution to the total.



Number of Y values/data point: 1

Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: Gap gets or sets the space between the bars of each X axis value

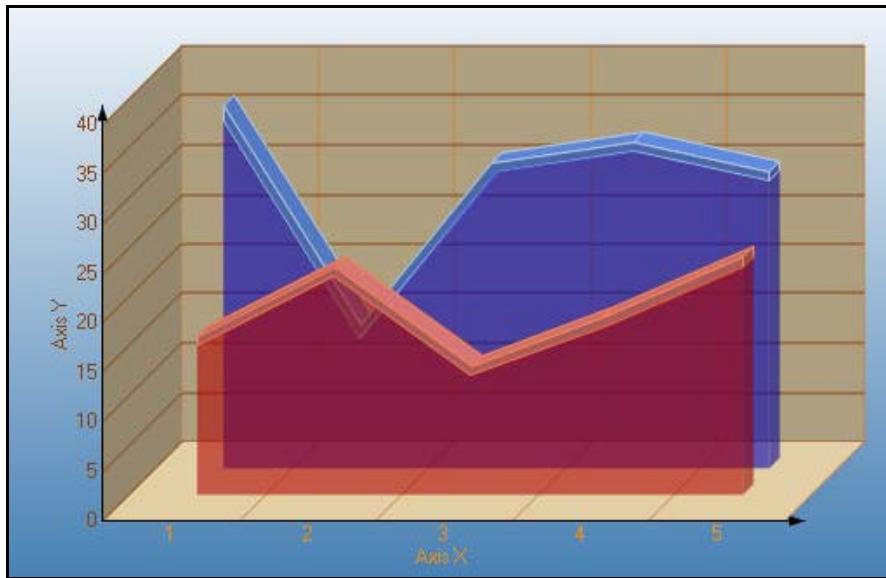
3D Charts

This topic illustrates some of the three dimensional chart types that you can create with the Chart control.

Note: To see a chart in three dimensions, open the ChartArea Collection dialog, and in the Projection section, change the ProjectionType from Identical to Orthogonal.

- **Area3D Charts**

Use a 3D area chart to compare trends in two or more data series over a period of time or in specific categories, allowing the data to be viewed side by side.



Number of Y values/data point: 1

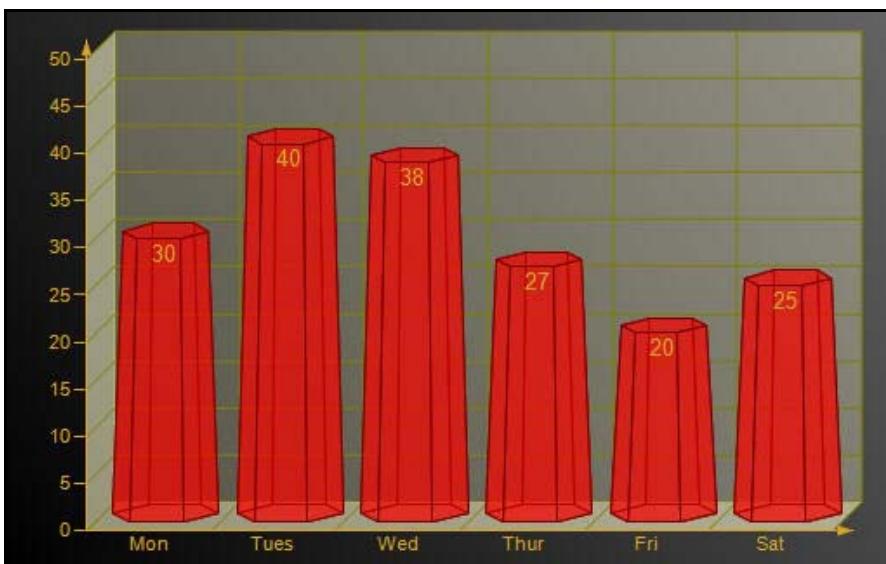
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: LineBackdrop gets or sets the backdrop information for the 3D line. Thickness gets or sets the thickness of the 3D line. Width gets or sets the width of the 3D line.

- **Bar3D Charts**

Use a 3D bar chart to compare values of items across categories, allowing the data to be viewed conveniently in a 3D format.



Number of Y values/data point: 1

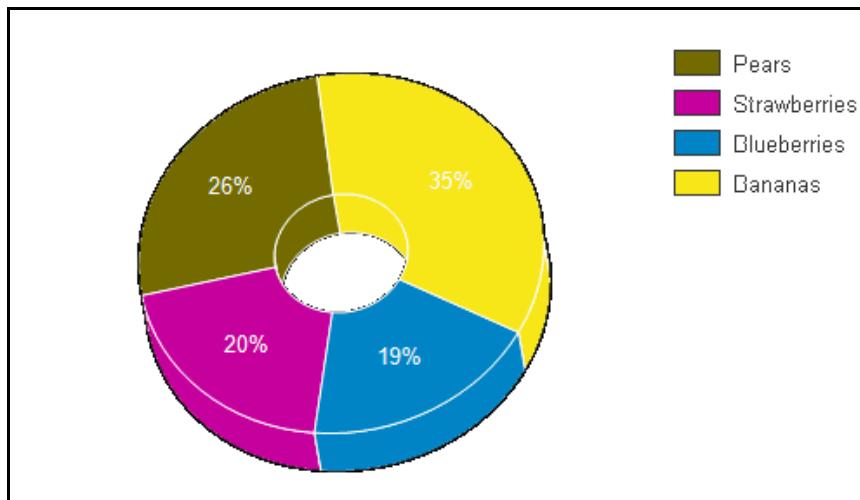
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: BarTopPercent gets or sets the percentage of the top of the bar that is shown for Cone or Custom BarTypes. BarType gets or sets the type of bars that is displayed. Gap gets or sets the space between the bars of each X axis value. RotationAngle gets or sets the starting horizontal angle for custom 3D bar shapes. Can only be used with the Custom BarType. VertexNumber gets or sets the number of vertices for the data point, used to create custom 3D bar shapes. Can only be used with the CustomBarType. Bars must contain 3 or more vertices.

- **Doughnut3D Pie Charts**

A 3D doughnut chart shows how the percentage of each data item contributes to a total percentage, allowing the data to be viewed in a 3D format.



A 3D doughnut chart shows how the percentage of each data item contributes to a total percentage, allowing the data to be viewed in a 3D format.

Number of Y values/data point: 1

Number of Series: 1

Marker Support: Series or Data Point

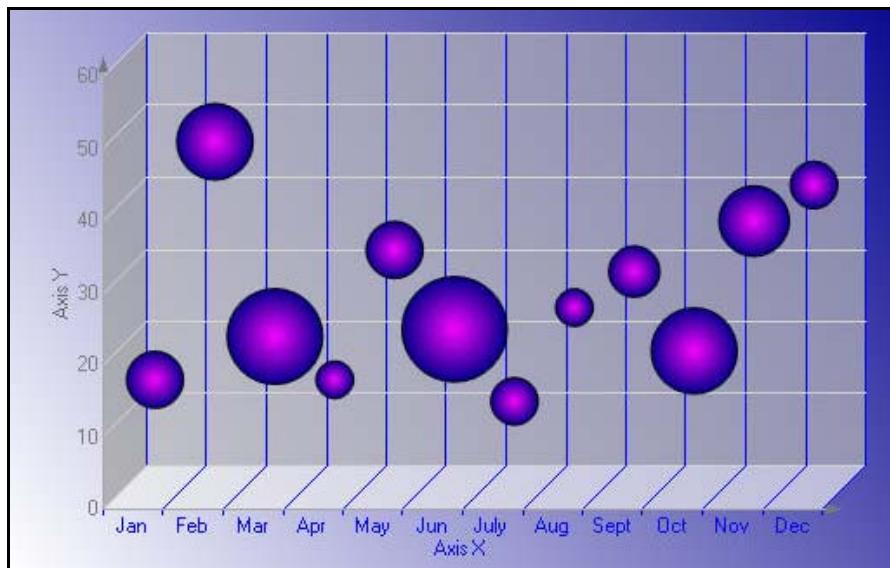
Custom Properties: ExplodeFactor gets or sets the amount of separation between data point values. The value must be less than or equal to 1. To explode one section of the doughnut chart, set ExplodeFactor on the data point instead of on the series. HoleSize gets or sets the inner radius of the chart. If set to 0, the chart will look like a pie chart. The value must be less than or equal to 1. OutsideLabels gets or sets a value indicating whether the data point labels appear outside of the graph. StartAngle gets or sets the horizontal start angle for the series data points.

XY Charts

Some of the XY chart types you can create with the Chart control are described below.

- **Bubble Charts**

The Bubble chart is an XY chart in which bubbles represent data points. The first Y value is used to plot the bubble along the Y axis, and the second Y value is used to set the size of the bubble. The bubble shape can be changed using the series Shape property.



Number of Y values/data point: 2

Number of Series: 1 or more

Marker Support: Series or Data Point. Marker labels use the second Y value as the default value.

Custom Properties: MaxSizeFactor gets or sets the maximum size of the bubble radius. Values must be less than or equal to 1. Default is .25. MaxValue gets or sets the bubble size that is used as the maximum. MinValue gets or sets the bubble size that is used as the minimum. Shape gets or sets the shape of the bubbles. Uses or returns a valid MarkerStyle enumeration value.

- **LineXY Charts**

A line XY chart plots points on the X and Y axes as one series and uses a line to connect points to each other.



Number of Y values/data point: 1

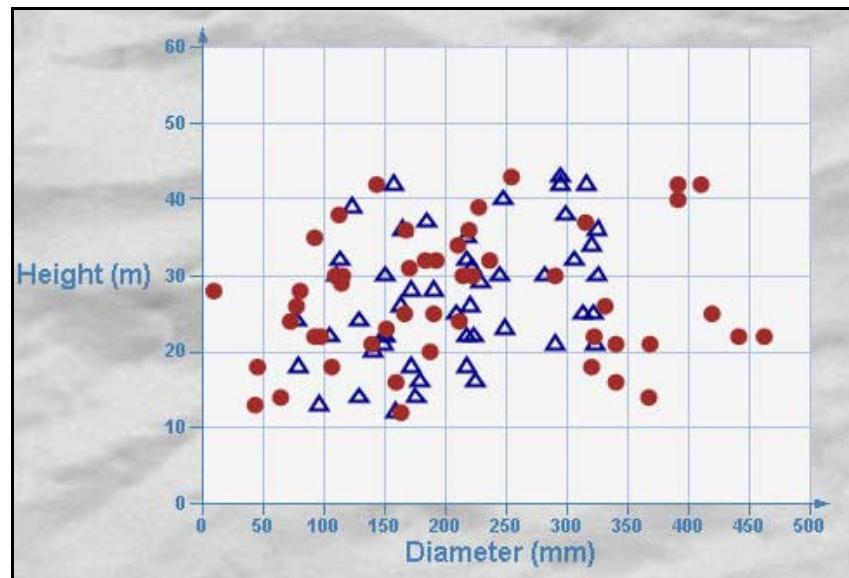
Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

- **PlotXY Charts**

A plot XY chart shows the relationships between numeric values in two or more series sets of XY values.



Number of Y values/data point: 1

Number of Series: 1 or more

Marker Support: Series or Data Point

Custom Properties: None

Chart Data

Data-Bound Charts

The Chart control provides several ways to bind your charts to data at design time.

- Adding Data with the Wizard

To open the Chart Wizard, right-click the chart and select Wizard. In the Chart Wizard, once you have added a series, you can create a data adapter to contain the data for your chart, if needed. When a data source is available, the Value X and Y values can be set for the series in the chart wizard from the expressions and/or data columns retrieved from the data source.

- Adding Data with the Chart Designer

Once a data source is set up, you can easily bind data to a series using the Chart Designer. To open the Chart Designer, click the Customize verb below the *Properties* window. Choose the Series section on the left, and on the General tab, after a series has been added to the chart, set the ValueY property by selecting the name of the data expression you wish to assign to the series.

- Adding Data through the *Chart Data Source* Dialog

To set the data source for the chart through the *Chart Data Source* dialog, click the DataSource property.

After the DataSource for the chart is set, add a series to the chart. To do this, open the *Series Collection Editor* dialog by clicking the browse button  which appears when you click next to the Series property in the *Properties* window, then click the **Add** button. To bind the series to an expression or dataset column returned by your data source, set the ValueMembersY or ValueMembersX property of the series by selecting it from the drop-down list.

Unbound Charts

The Chart control makes it easy to set the data source for a chart control, series, or data points collection at run time.

Below is a list of objects that can be used as data sources.

- dataset
- dataset Column
- Data Table
- SqlCommand/OleDbCommand
- SqlDataAdapter/OleDbDataAdapter
- Array

Below are some examples of binding to different data sources at run time.

dataset

The Chart control's DataSource property can be set to a dataset at run time. The following code demonstrates setting up a dataset, setting the DataSource property to the dataset, creating a series, and setting the ValueMembersY property to the dataset expression at run time.

```

// C#
// create the series
DataDynamics.ActiveReports.Chart.Series s = new
DataDynamics.ActiveReports.Chart.Series();
string m_cnnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:/Northwind.mdb;Persist
Security Info=False";
System.Data.OleDb.OleDbConnection m_cnn = new
System.Data.OleDb.OleDbConnection(m_cnnString);
System.Data.OleDb.OleDbDataAdapter oDBAdapter;
// create the dataset
System.Data.DataSet oDS;
oDBAdapter = new System.Data.OleDb.OleDbDataAdapter("SELECT ShipCountry,
SUM(Freight) AS
Expr1 FROM Orders GROUP BY ShipCountry", m_cnnString);
oDS = new System.Data.DataSet();
oDBAdapter.Fill(oDS, "Expr1");
// set the DataSource and ValueMembersY properties
this.ChartControl1.DataSource = oDS;
s.ValueMembersY = "Expr1";
this.ChartControl1.Series.Add(s);

```

dataset Column

In the Chart control, the ValueMembersX and ValueMembersY properties of a series can be set to a dataset column. The following code demonstrates creating a series, setting up a dataset, setting the DataSource property to the dataset, and setting the ValueMembersY and ValueMembersX properties to dataset columns at run time.

```

// C#
// create the series
DataDynamics.ActiveReports.Chart.Series s = new
DataDynamics.ActiveReports.Chart.Series();
string m_cnnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:/Northwind.mdb;Persist
Security Info=False";
System.Data.OleDb.OleDbConnection m_cnn = new
System.Data.OleDb.OleDbConnection(m_cnnString);
System.Data.OleDb.OleDbDataAdapter oDBAdapter;
// create the dataset
System.Data.DataSet oDS;
oDBAdapter = new System.Data.OleDb.OleDbDataAdapter("SELECT * from Orders
WHERE OrderDate
< #08/17/1994#", m_cnnString);

```

```

oDS = new System.Data.DataSet();
oDBAdapter.Fill(oDS, "Orders");
// set the DataSource, ValueMembersY, and ValueMembersX properties
this.ChartControl1.DataSource = oDS;
this.ChartControl1.Series.Add(s);
this.ChartControl1.Series[0].ValueMembersY =
oDS.Tables["Orders"].Columns[7].ColumnName;
this.ChartControl1.Series[0].ValueMemberX =
oDS.Tables["Orders"].Columns[8].ColumnName;

```

Data Command

A chart's data source can be set to a SqlCommand or OleDbCommand. The following code demonstrates creating a series, creating an OleDbCommand, setting the DataSource property to the data command, and setting the ValueMembersY property for the series at run time.

```

// C#
// create the series
DataDynamics.ActiveReports.Chart.Series s = new
DataDynamics.ActiveReports.Chart.Series();
string m_cnnString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:/"
Northwind.mdb;Persist
    Security Info=False";
System.Data.OleDb.OleDbConnection m_cnn = new
System.Data.OleDb.OleDbConnection(m_cnnString);
string query = "SELECT ShipCountry, SUM(Freight) AS Expr1 FROM Orders GROUP
BY ShipCountry";
// create the OleDbCommand and open the connection
System.Data.OleDb.OleDbCommand command = new
System.Data.OleDb.OleDbCommand(query, m_cnn);
command.Connection.Open();
// set the DataSource and ValueMembersY properties
this.ChartControl1.DataSource = command;
this.ChartControl1.Series.Add(s);
this.ChartControl1.Series[0].ValueMembersY = "Expr1";
// close the connection
m_cnn.Close();

```

Array

The Chart control allows the data source for the data points collection to be set to an array. The following code demonstrates creating a series, creating an array, and using the DataBindY method to set the data source for the data points collection at run time.

```

// C#
// create the series

```

```

DataDynamics.ActiveReports.Chart.Series s = new
DataDynamics.ActiveReports.Chart.Series();
// create the array
double [] a = {1,4,2,6,3,3,4,7};
// set the data source for the data points collection
this.ChartControl1.Series.Add(s);
this.ChartControl1.Series[0].Points.DataBindY(a);

```

Calculated and Sequence Series Charts

The Chart control allows you to bind a formula to the ValueMembersY property of a series to create a calculated or sequence series for your chart.

Calculated Series

You can easily create a calculated series based on the values of one or more series by setting the ValueMembersY property of a series to a formula. To reference a series in the formula, use the name of the series. The following code demonstrates creating two series, one bound to a data array and the other bound to a formula based on the Y values of the first series.

// C#

```

DataDynamics.ActiveReports.Chart.Series s = new
DataDynamics.ActiveReports.Chart.Series();
DataDynamics.ActiveReports.Chart.Series cS = new
DataDynamics.ActiveReports.Chart.Series();
double [] a = { 1,4,2,6,3,3,4,7};
this.ChartControl1.Series.AddRange(new DataDynamics.SharpGraph.Windows.Series[]
{s, cS});
this.ChartControl1.Series[0].Name = "Series1";
this.ChartControl1.Series[0].Points.DataBindY(a);
this.ChartControl1.Series[1].ValueMembersY = "Series1.Y[0]+10";

```

Sequence Series

Set a sequence series by specifying the minimum value, maximum value, and step for the series. The following code shows how to set the ValueMembersY property at run time to create a sequence series.

// C#

```

DataDynamics.ActiveReports.Chart.Series s = new
DataDynamics.ActiveReports.Chart.Series();
this.ChartControl1.Series.Add(s);
this.ChartControl1.Series[0].ValueMembersY = "sequence(12,48,4)";

```

Chart Effects

Colors

In the Chart control, colors can be used in different ways to enhance the chart's appearance, distinguish different series, point out or draw attention to data information such as averages, and more.

Color Palettes

The Chart control includes several pre-defined color palettes that can be used to automatically set the colors for data values in a series. The pre-defined palettes are as follows:

- Cascade (default): A cascade of eight cool colors ranging from deep teal down through pale orchid.
- Confetti: A sprinkling of bright and pastel colors.
- Iceberg: A range of the soft blues and greys found in an iceberg.
- Springtime: The colors of spring, in deep green, two vivid colors and five pastels.
- None: All data is drawn using the same teal color.

These enumerated values are accessed through the Series class with code like the following.

```
// C#
this.ChartControl1.Series[0].ColorPalette = DataDynamics.ActiveReports.Chart.
ColorPalette.Iceburg;
```

Gradients

Gradients can be used in object backdrops to enhance the visual appearance of various chart items. Gradients can be used in the following chart sections:

- Chart backdrop
- Chart area backdrops
- Wall backdrops
- Title backdrops
- Legend backdrops
- Legend item backdrops (for custom legend items)
- WallRange backdrops
- Series backdrops
- Data point backdrops
- Marker backdrops
- Marker label backdrops
- Annotation TextBar backdrops

3D Effects

Using the projection and viewpoint settings, you have the ability to display your 3D chart at or from any angle needed to provide the desired view or call attention to a specific chart section.

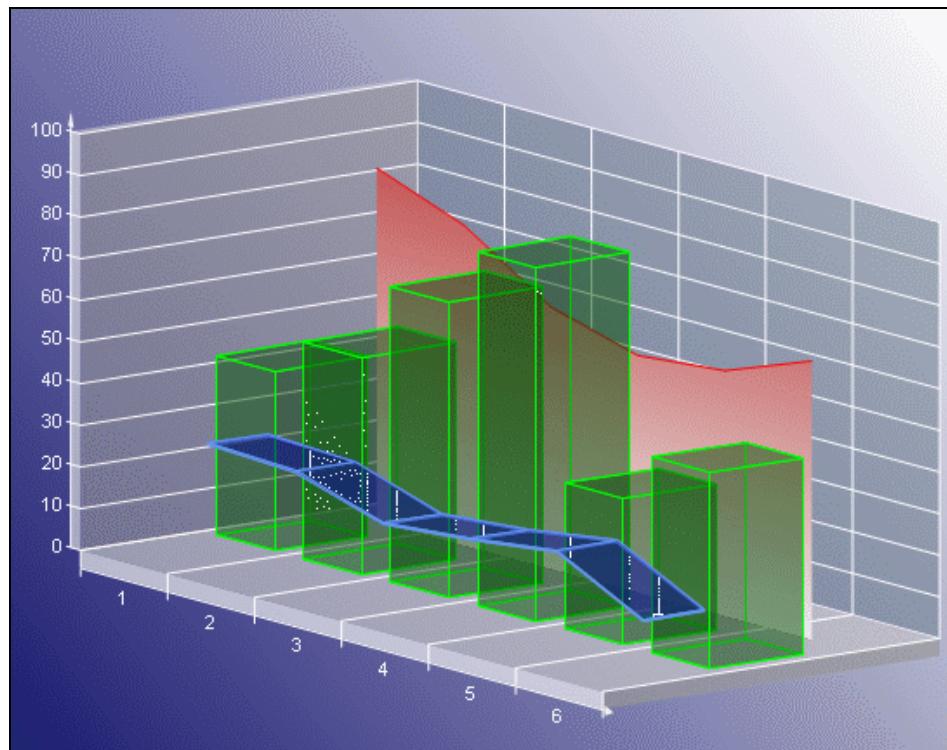
Projection

Determine the projection for a 3D chart using three factors: the ZDepth ratio, the projection type, and the projection DX and DY values.

- **ZDepth ratio** The Z depth ratio is the level of depth the Z axis has in the chart. Values range from 0 (for a 2D chart) to 1.0.
- **ProjectionType** The type of projection used for the chart. In order to show charts three dimensionally, the ProjectionType in the ChartArea Collection editor must be set to Orthogonal. To access this dialog box, click the browse button  next to the ChartAreas (Collection) property in the *Properties* window.
- **ProjectionDX** The origin position of the Z axis in relation to the X axis. This property is valid only when the ProjectionType is Orthogonal.
- **ProjectionDY** The origin position of the Z axis in relation to the Y axis. This property is valid only when the ProjectionType is Orthogonal.
- **HorizontalRotation** The HorizontalRotation property allows you to set the degree (-90° to 90°) of horizontal rotation from which the chart is seen.
- **VerticalRotation** The VerticalRotation property allows you to set the degree (-90° to 90°) of vertical rotation from which the chart is seen.

Lighting

The Chart control provides the ability to completely customize lighting options for 3D charts.



Directional Light Ratio

Using the DirectionalLightRatio property, you can control the directional or ambient intensity ratio.

Light Type

By setting the Type property to one of the enumerated LightType values, you can control the type of lighting used in the chart. The settings are as follows:

- Ambient An ambient light source is used. It is equal to DirectionalLightRatio = 0.
- InfiniteDirectional An infinite directional light source (like the sun) is used.
- FiniteDirectional A point light source is used.

Light Source

You can also set the Source property to a Point3d object, which controls the location of the light source.

Alpha Blending

The Backdrop class in the Chart control has an Alpha property which employs GDI+, and is used to set the transparency level of each object's backdrop. GDI+ uses 32 bits overall and 8 bits per alpha, red, green, and blue channels respectively to indicate the transparency and color of an object. Like a color channel's levels of color, the alpha channel represents 256 levels of transparency.

The default value of the Alpha property is 255, which represents a fully opaque color. For a fully transparent color, set this value to 0. To blend the color of the object's backdrop with the background color, use a setting between 0 and 255.

In the Chart control, you can use the Color.FromArgb method to set the alpha and color levels for a particular chart element. The following example shows how you can use the method to set the alpha and color values for the chart backdrop.

// C#

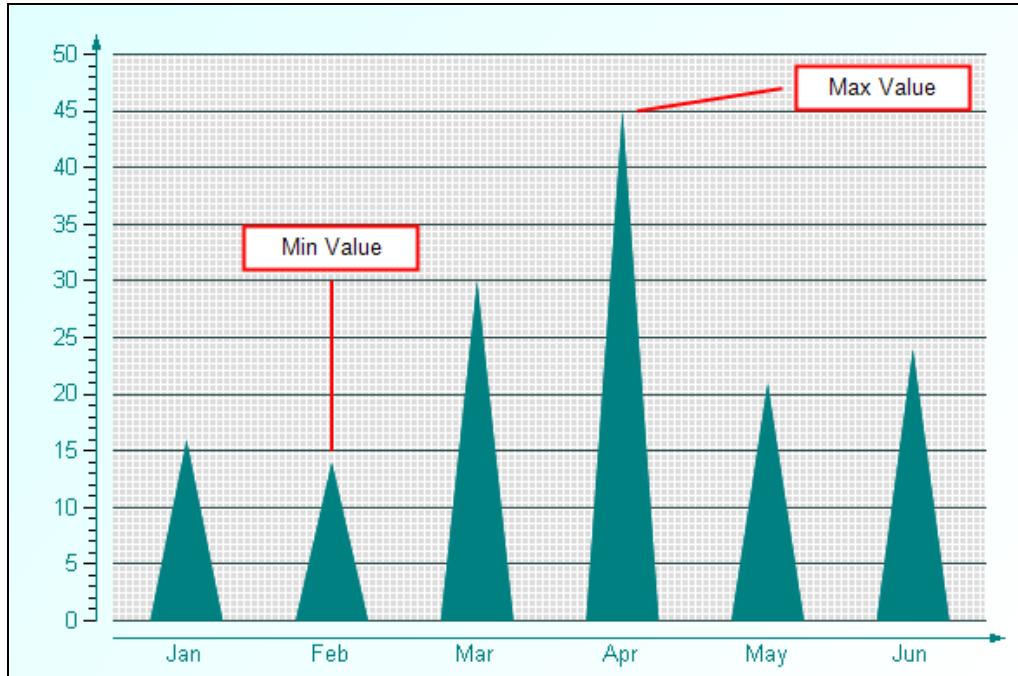
```
this.ChartControl1.Backdrop = new DataDynamics.ActiveReports.Chart.  
BackdropItem(Color.FromArgb(100, 0, 11, 220));
```

Changing the alpha level of a chart element reveals other items that are beneath the object. Because you can set the alpha level for any chart element that supports color, you can create custom effects for any chart. For example, you can use alpha blending to combine background images with a semi-transparent chart backdrop to create a watermark look.

Chart Control Items

Annotations

The Chart control offers a built-in annotation tool to allow you to include floating text bars or images in your charts or call attention to specific items or values in your charts using the line and text bar controls included in the Annotation Collection Editor.

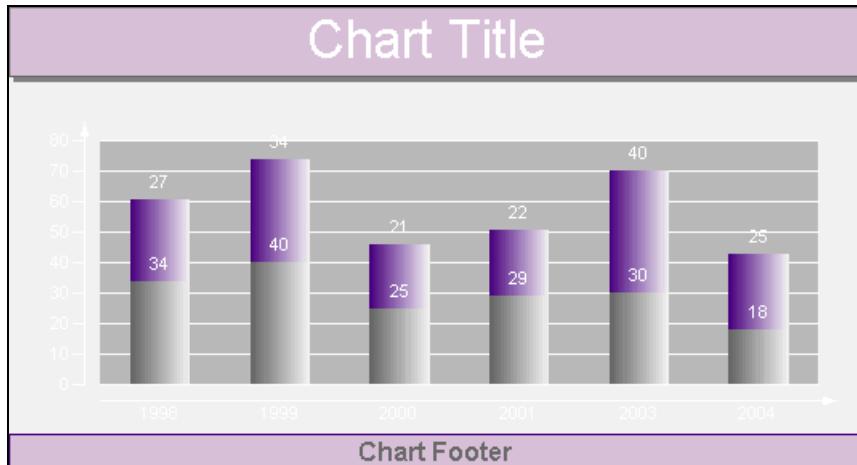


The following properties are important when setting up annotations for your chart:

- Start Point: sets the starting point (X and Y axis values) for an annotation line.
- End Point: sets the end point (X and Y axis values) for an annotation line.
- Anchor Placement: sets the position of the anchor point for the text bar on the chart surface.
- Anchor Point: sets the point (X and Y axis values) where the text bar will be anchored based on the anchor placement selected.

Titles and Footers

The Chart control allows you to add custom titles to your charts. The Titles collection is accessible from the SharpGraph object. With the ability to add as many titles as needed, dock them to any side of a chart area, change all of the font properties, add borders and shadows, make the background look the way you want it, and change the location of the text, you can easily make your titles look the way you want them to look.

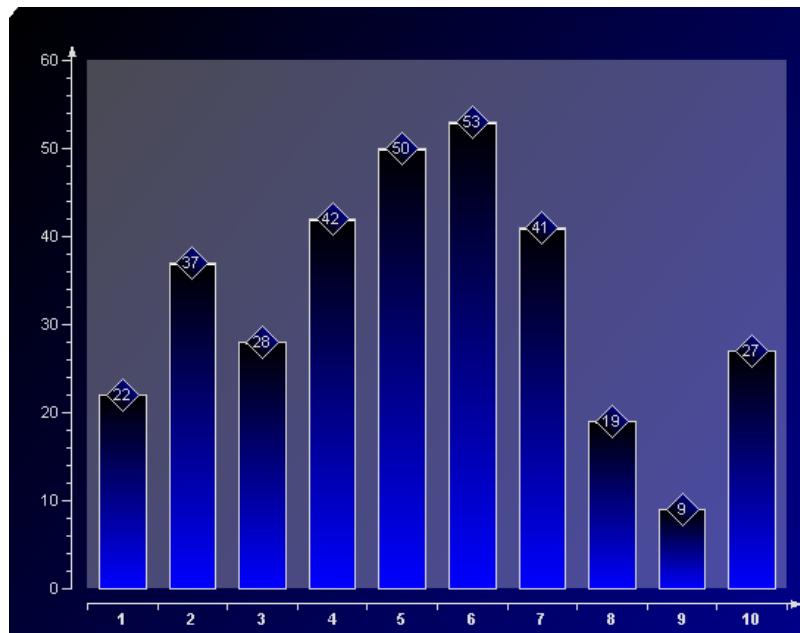


Legends

The Chart control automatically creates a legend item for each series added to a chart at design time and sets the Legend property for each series by default. However, the legend's Visible property must be set to True for the legend to show with the chart. The text for each default legend entry is taken from the Name property on the series. Each Series to be shown in the Legend must have a Name. If the Name property is not set, the Series does not show up in the Legend.

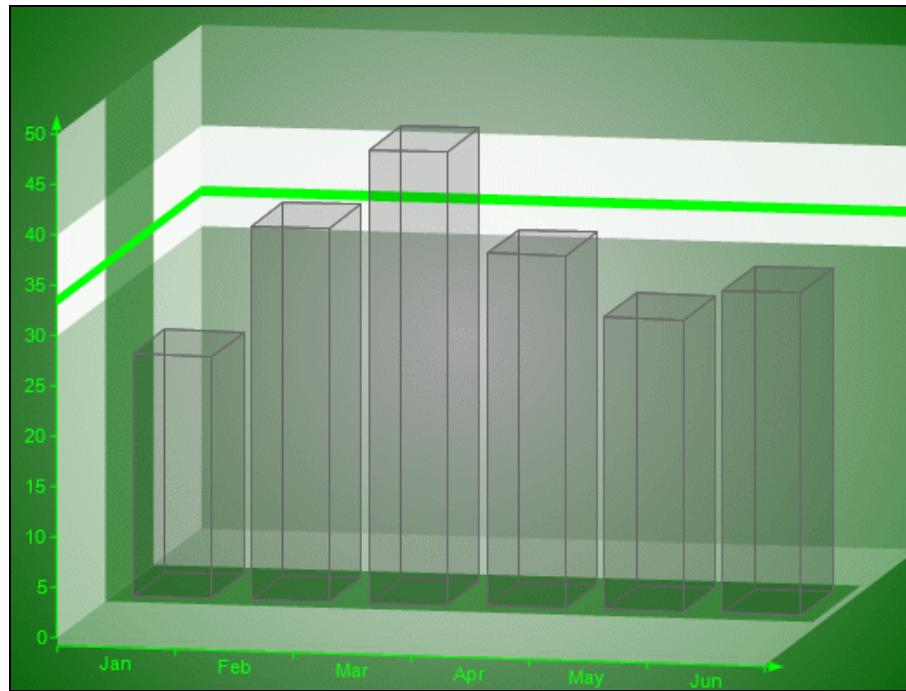
Markers

Markers are used to show specific data series values in a chart.



Constant Lines and Stripes

The Chart control supports constant lines and stripes through the use of the WallRanges collection. It allows you to display horizontal or vertical lines or stripes in a chart to highlight certain areas. For example, you could draw a stripe in a chart to draw attention to a high level in the data or draw a line to show the average value of the data presented.



Important properties

- EndValue--Sets the end value on the primary axis for the wall range.
- StartValue --Sets the start value on the primary axis for the wall range.
- PrimaryAxis--Sets the axis on which the wall range should appear.

Chart Axes and Walls

Standard Axes

The Chart control provides the means to change axis settings at design time or run time. Chart axes make it possible to view and understand the data plotted in a graph.

Axis Types

Most 2D charts contain a numerical axis (AxisY) and a categorical axis (AxisX). 3D charts include another numerical axis (AxisZ). These axes are accessible at run time from the ChartArea object and allow you to control the settings for each, including scaling, labels, and various formatting properties. For any of the scaling or labeling properties you set to show up at run time, you will need to set the Visible property of the axis to True.

Changing Axis Settings

Axis settings can be changed at design time by clicking on a Chart control and using the *Properties* window or at run time in code from the chart's ChartArea object.

Scaling

For normal linear scaling on a numeric axis, you will need to set the Max and Min properties for the axis, which correspond to the numerical values in the chart's data series. You will also need to set the Step property of the MajorTick to show the major numerical unit values. The Step property controls where labels and/or tick marks are shown on the numerical axis.

// C#

```
this.ChartControl1.ChartAreas[0].Axes["AxisY"].Max = 100;  
this.ChartControl1.ChartAreas[0].Axes["AxisY"].Min = 0;  
this.ChartControl1.ChartAreas[0].Axes["AxisY"].MajorTick.Step = 10;
```

The Chart control also supports logarithmic scaling which allows you to show the vertical spacing between two points that corresponds to the percentage of change between those numbers. You can set your numeric axis to scale logarithmically by setting the IsLogarithmic property on the axis to True and setting the Max and Min properties of the axis.

Labeling

To show labels on an axis, you will need to specify the value for the LabelsGap property, set your LabelsFont properties, and set LabelsVisible to True. These properties can be set in the AxisBase Collection editor, which is accessed at design time by clicking the browse button  next to the ChartAreas (Collection) property, then the Axes (Collection) property of the ChartArea.

NOTE: Labels render first, and then the chart fills in the remaining area, so be sure to make the chart large enough if you use angled labels.

You can specify strings to be used for the labels instead of numerical values on an axis by using the Labels collection property at design time or assigning a string array to the Labels property at run time. You can also specify whether you want your axis labels to appear on the outside or inside of the axis line using the LabelsInside property. By default, labels appear outside the axis line.

Secondary Axes

By default, a Chart object includes secondary X and Y axes (AxisX2 and AxisY2). At design time or run time, you can specify a secondary axis to plot data against by setting all of the appropriate properties for AxisX2 or AxisY2, including the Visible property.

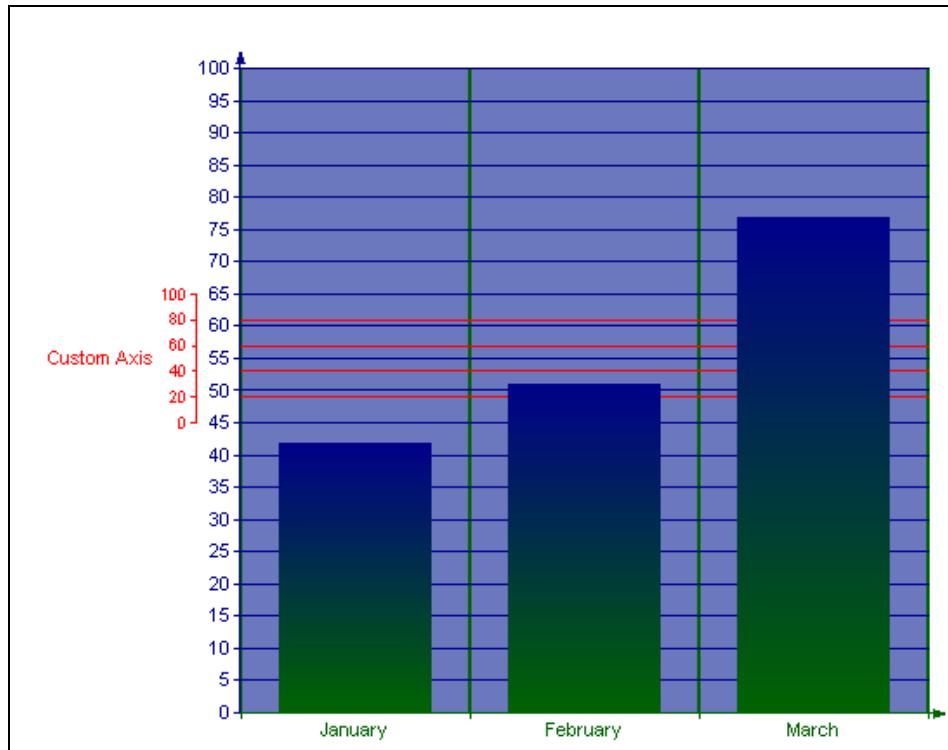
If you want to use two axes to show the same data as it appears on two different scales, you can set the primary axis to show the actual data value scale, for example, and set the secondary axis to show a logarithmic scale.

Custom Axes

The Chart control supports the creation of additional custom axes through the use of the chart's CustomAxes collection. Once a custom axis has been added to the collection, in addition to setting the normal axis properties, you will need to set the following properties:

- Parent—The Parent property allows you to choose the primary or secondary axis on which your custom axis resides.
- PlacementLength—The PlacementLength property allows you to set the length of the custom axis in proportion to the Min and Max property values you have already set for the parent axis.

- **PlacementLocation**—The PlacementLocation property allows you to set the starting location value for the custom axis to appear in relation to the parent axis.



Gridlines and Tick Marks

Gridlines and tick marks are generally used to help increase the readability of a chart.

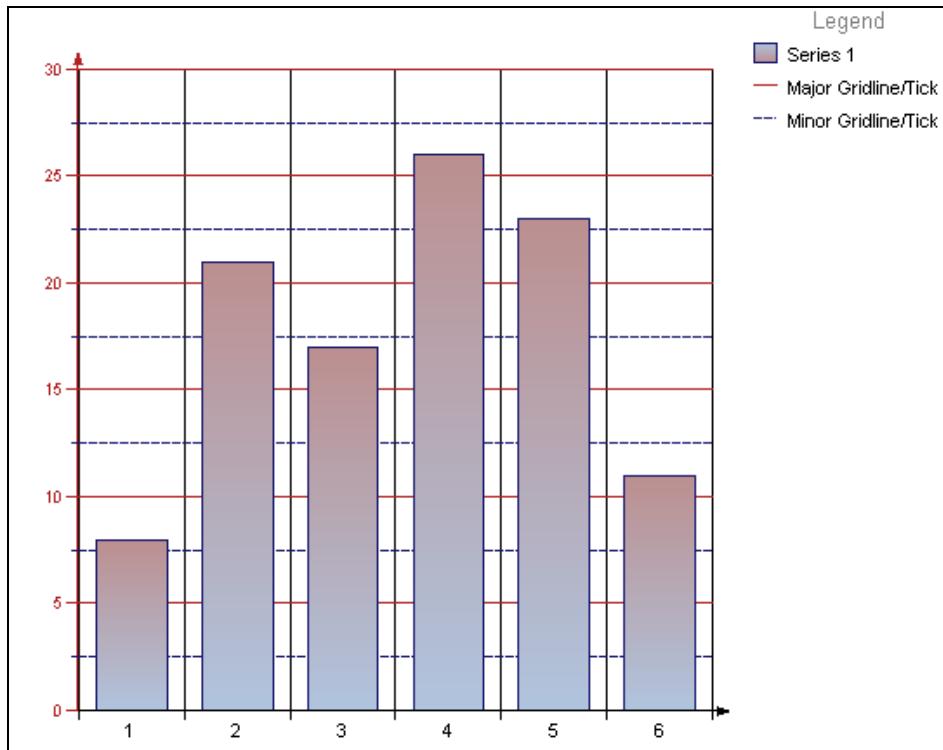


Chart-Specific Properties

Each chart type in the Chart control contains specific properties that apply to it. Set the chart type and chart-specific properties in the *Series Collection Editor* dialog box accessed through the Series property in the property grid and in the *DataPoint Collection* dialog box accessed through the Points property in the *Series* dialog box.

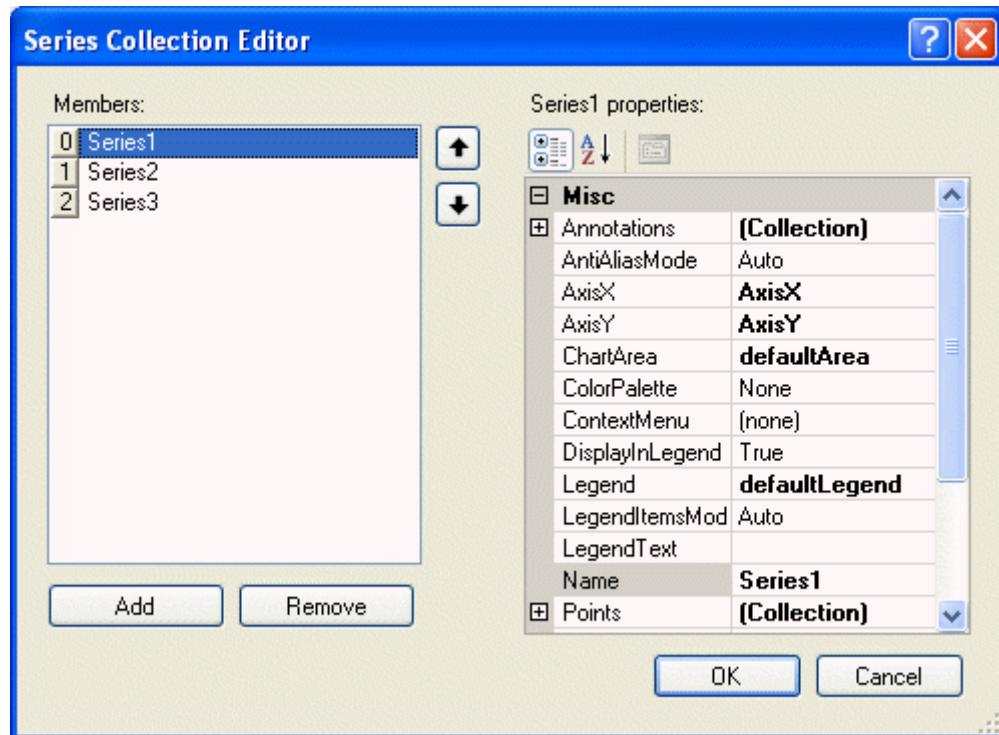
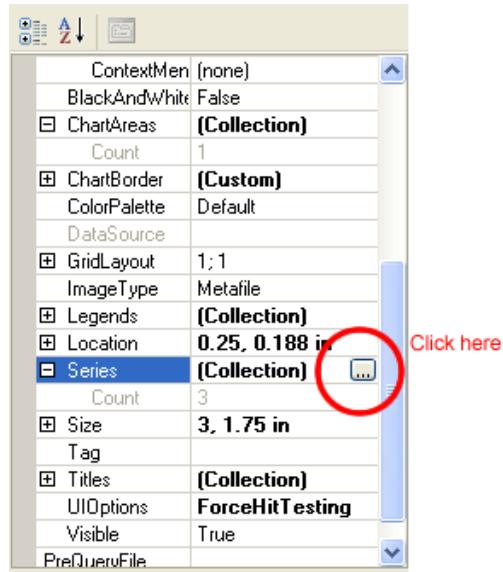
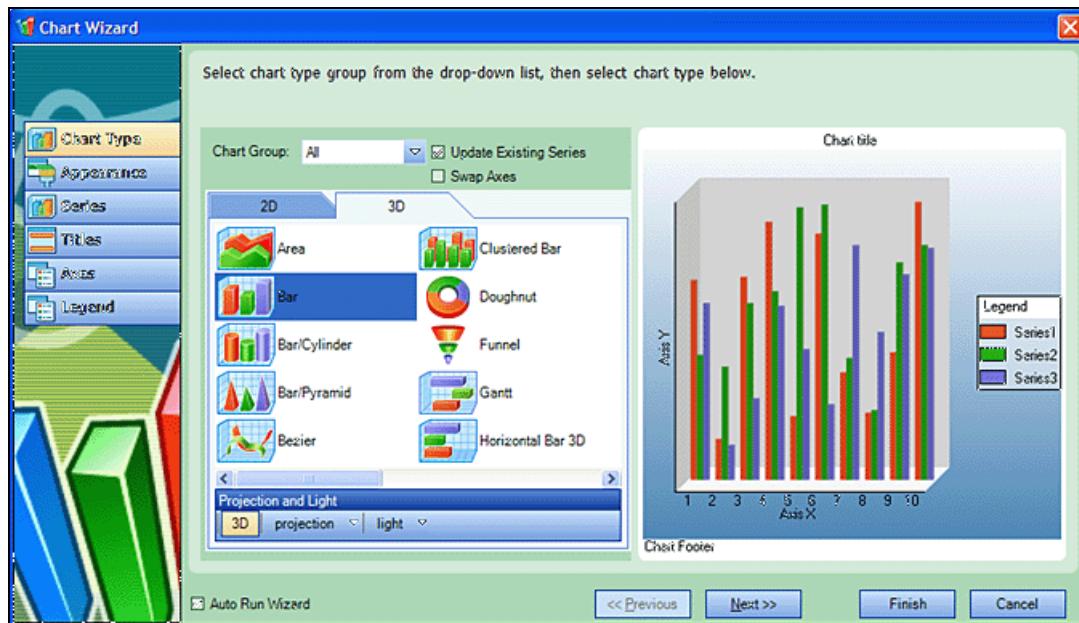


Chart Wizard

The chart control features an easy-to-use wizard. The chart wizard automatically runs when you first add a chart control to a report. If you prefer not to have the wizard run automatically, uncheck the Auto Run Wizard check box at the bottom of the wizard. You can also access the wizard through the Wizard verb that appears below the *Properties* window when the chart is selected on the report.



Walk-Through: Creating a Report

In this exercise, you will build a report similar to the vulnerability (classic) report, but not as intricate.

Task 1: Build the master report

- 1 Click **File → New**.
- 2 On the *Create Report Definition* window, enter the following:
 - Name: My vulnerability report
 - Description: Sample report
- 3 In the Context list, select **Scan**.
- 4 Select **Exposed in Product**.
- 5 In the **View Name** list, select **Basic - Server Information**.
- 6 Click **OK**.
- 7 Right-click the PageHeader caption and select **Delete**.
- 8 Right-click the Detail caption and select **Insert → Report Header/Footer**.
- 9 In the toolbox, drag **LinkedSubReportControl** into the **ReportHeader** section.
- 10 On the *Choose a Report* dialog, expand the **Vulnerability (classic)** group, select **ScanHeader**, and click **OK**.

- 11 Position the element and extend it to the right margin.
- 12 Click the ReportHeader caption.
- 13 In the Properties grid, set CanShrink = True.
- 14 Click the Detail caption.
- 15 In the Properties grid, set CanShrink = True.

Task 2: [Add a Link to a Subreport](#)

- 1 In the toolbox, drag a LinkedSubReportControl into the Detail section.
- 2 On the *Choose a Report* dialog, expand the Vulnerability (classic) group, select **ServerHeader**, and click **OK**.
- 3 Position the element and extend it to the right margin.
- 4 With the ServerHeader selected, in the Properties grid under Associated Fields, click **@ServerID** and select ServerID.
- 5 Click the **Preview** tab.
- 6 When prompted to design parameters, select **No**.
- 7 Select a scan and click **Next**.
- 8 When prompted to select a report, click **Finish**.
- 9 Click **File → Save**.

Task 3: [Create a Subreport](#)

- 1 Click **File → New**.
- 2 On the *Create Report Definition* window, enter or select the following:
 - a For the Name, enter “My vulnerability by server.”
 - b For the Description, enter “Sample report.”
- 3 From the **Context** list, select **Scan**.
- 4 Clear the **Exposed in Product** check mark.
- 5 In the **View Name** list, select *Basic - Vulnerability by Session*.
- 6 Click **OK**.
- 7 Delete the PageHeader caption (right-click the caption and select **Delete**).
- 8 In the Properties grid, set CanShrink = True.
- 9 Right-click the Detail caption and select **Insert → Group Header/Footer**.
- 10 In the Properties grid:
 - a Set CanShrink = True.
 - b Change the name to **GroupServer**.
 - c For the DataField, select **Server**.
- 11 Drag a BookmarkControl to the **GroupServer** area.
- 12 In the Properties grid, select **BookmarkText** and enter the following:
`{=MainReportName}\{=Server}`

Task 4: Add a chart to the report

- 1 Click **Edit** → **Modify/Create Report**.
- 2 Select **Aggregate - Severity Summary by Server** and click **OK**. This query will be used to generate a chart.
- 3 Drag a ChartControl onto the design area.
- 4 On the Chart Wizard, click the **2D** tab and select **Bar**.
- 5 Click **Finish**.
- 6 Resize the chart and arrange it to your liking.
- 7 With the chart selected, go to the Properties grid, click **AssociatedQuery**, and select the query you just added: Aggregate - Severity Summary by Server.
- 8 Right-click the chart and select **Wizard**.
- 9 Select **Series** from the list in the left-hand pane.
- 10 Assign a series to each severity category: critical, high, medium, low, informational, and best practice.
 - a Select **Series1**, and in the Series Properties area and enter “Critical” for the Name.
 - b In the Data Binding area, select the Y axis and select **Critical** from the drop-down list.
 - c Repeat this process for each series; click **Add New Item** where necessary.
- 11 Click **Finish**.
- 12 With the chart selected, go to the Properties grid and click **@ServerID** under AssociatedFields and select VulnerabilityCount.

Task 5: Add a section for the Check ID, Check Severity, and Check Name, and Summary

- 1 Right-click the Detail caption and select **Insert** → **Group Header/Footer**.
- 2 Collapse the footer.
- 3 Click the GroupHeader.
- 4 In the Properties grid:
 - a Change the name to “groupCheck.”
 - b Set **CanShrink** = True
 - c For the DataField, select “checkid.”
- 5 Drag a TextBox to the groupCheck section.
- 6 In the Properties grid:
 - a For Name, enter txtSeverity
 - b For DataField, select “checkseverity.”
- 7 Drag another TextBox into the groupCheck section and place it to the right of the first TextBox.
- 8 In the Properties grid:
 - a Change the name to “txtCheckName.”
 - b Set **CanShrink** = True
 - c For the DataField, select “checkname.”

- d For ClassName, select Normal Bold.
- 9 Drag a Label into the area.
- 10 In the Properties grid:
 - a Change the name to lblSummary.
 - b For Text, enter Summary.
- 11 Drag a RichTextBox onto the canvas; place it below the summary label and extend it to the right.
- 12 In the Properties grid:
 - a Change the name to txtSummary.
 - b For the DataField, select ReportSection_Summary.
- 13 Drag a BookmarkControl and place it anywhere on the groupCheck canvas
- 14 On the Properties grid:
 - a For BookMarkText, enter {=MainReportName}\Checks\{=Checkid}.
 - b For the Name, enter BookmarkChecks.

Task 6: [Add an area for the HTTP Request](#)

- 1 Right-click on the Detail caption and select **Insert → Group Header/Footer**.
- 2 Collapse the group footer.
- 3 On the Properties grid:
 - a Set CanShrink =True.
 - b For the Name, enter groupRequest.
- 4 Drag a RichTextBox to the groupRequest canvas. Size it and extend it to the right margin.
- 5 On the Properties grid:
 - a Set CanShrink =True.
 - b For the Name, enter txtRequest.
 - c For the DataField, select RequestText.
 - d For TruncateVulnerability, select True.
 - e For HighlightVulnerability, select True

Task 7: [Add an area for the HTTP Response](#)

- 1 Drag a RichTextBox to the groupRequest canvas. Size it and extend it to the right margin.
- 2 On the Properties grid:
 - a Set CanShrink =True.
 - b For the Name, enter txtResponse.
 - c For the DataField, select ResponseText.
 - d For TruncateVulnerability, select True.
 - e For HighlightVulnerability, select True

Populate the Detail section

- 1 Drag the bound field “fullURL” to the Detail section.
- 2 Click the Parameter Designer icon on the toolbar.
- 3 In the Parameter Designer Canvas area, delete all parameters (click in the area, press **Ctrl + a**, and then press **Delete**).
- 4 Click **Save and Close**.

Task 8: Add/Modify the script

- 1 Click the **Script** tab on the Report Designer.
- 2 Change the method name “myEventHandler” to “onGroupCheckFormat.”
- 3 Delete all the script and replace with the following:

```
using System;
using DataDynamics.ActiveReports;
using HP.AppSec.Reporting.ReportScript;
namespace Script.Events
{
    public class MyEventClass
    {
        /*
         * You can declare fields, events and methods just like in c#...
         * in fact this is C!
         */
        /*
         * Script event handlers, MUST have this method signature
         */
        public void OnGroupCheckFormat(ReportObject report, EventArgs ea)
        {
            int nSeverity = (int)report.Fields["checkseverity"];
            TextBox txtSeverity =
report.CurrentSection.Controls["txtSeverity"] as TextBox;
            if (nSeverity <= 10)
            {
                txtSeverity.Text = "Informational";
            }
            else if( 10 < nSeverity && nSeverity <= 25)
            {
                txtSeverity.Text = "Low";
            }
            else if( 25 < nSeverity && nSeverity <= 50)
            {
                txtSeverity.Text = "Medium";
            }
            else if( 50 < nSeverity && nSeverity <= 75)
            {
                txtSeverity.Text = "High";
            }
            else if( 75 < nSeverity && nSeverity <= 100)
            {
```

```
        txtSeverity.Text = "Critical";
    }
}
}
```

- 4 After entering the script, click the **Report Events** tab (in the lower right) and select **groupCheck** from the drop-down list.
 - 5 For the Section Format Event, select `Script.Events.MyEventClass.onGroupCheckFormat`.
 - 6 Save the report.

Task 9: Add a pre-query to the master report

- 1 Open MyVulnerability report (listed under Custom Reports on the *Open a Report* dialog).
 - 2 Click **Edit** → **Modify/Create Report**.
 - 3 From the **View Name** list, select **PreQuery - Vulnerability**.

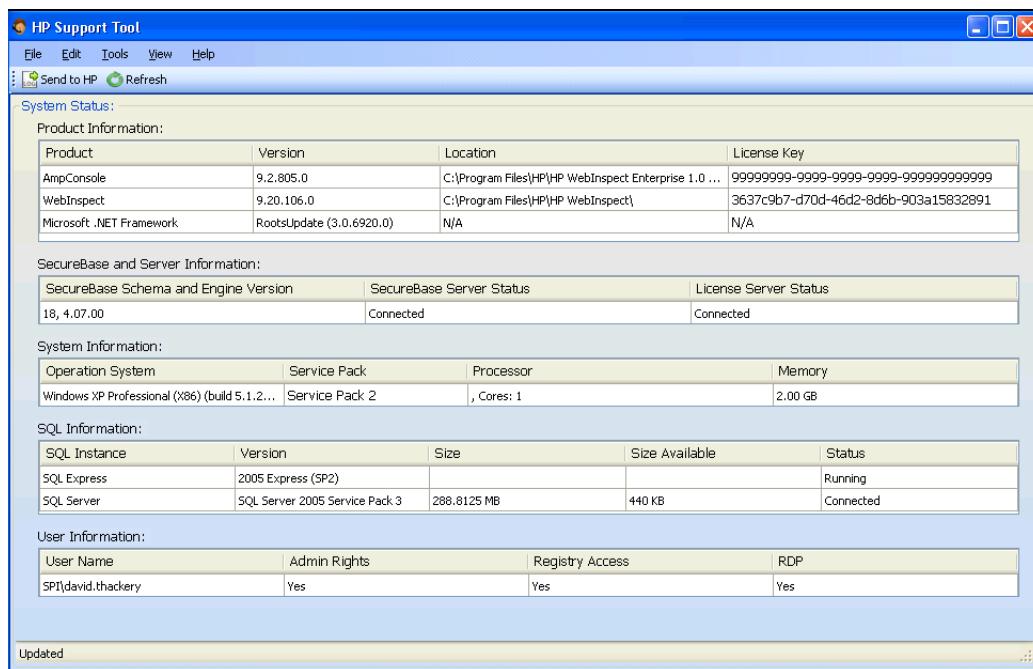
A pre-query improves performance by first determining if any data is available for the report.
 - 4 Drag a **LinkedSubReportControl** onto the Detail area.
 - 5 From the *Choose a Report* dialog, select My vulnerability by server and click **OK**.
 - 6 Position the control and extend it to the right margin.
 - 7 On the Properties grid:
 - a Under AssociatedFields, click **@serverID** and select serverID.
 - b For PreQueryFile, select PreQuery - Vulnerability.
 - 8 Click **Save**.
 - 9 Click the **Preview** tab.
 - 10 Note and correct any improperly positioned controls, then save your work.

HP Support Tool

The HP Support tool provides a quick and simple method for uploading files that may help HP support personnel to analyze and resolve any problems you encounter while using Application Security Center products. All communication uses Secure Sockets Layer (SSL) or FTP Secure (FTPS) protocol.

To launch the Support tool, click **Start** → **All Programs** → **HP** → **HP Software Tools** → **HP Support Tool**.

When first opened, the Support Tool displays information about ASC products and related system components. If data is not displayed, click **Refresh**.



Before sending data to HP, you may want to perform the following functions, which are available from the Tools menu.

- **Refresh WebInspect scans** - Deletes the scans.xml file and regenerates it, thereby refreshing the list of WebInspect scans that are displayed on the Manage Scans page. This is intended for use if scans are not displaying properly. Use this function only if directed by HP Support personnel.
- **Restore WebInspect SecureBase** - Replaces the current SecureBase with the factory default version. You can replace the Main SecureBase, the Scheduler SecureBase, or both. Use this function only if directed by HP Support personnel.
- **Check Services** - Displays a list of services associated with Application Security Center (ASC) products, allowing you to start, stop, or restart the services you select.

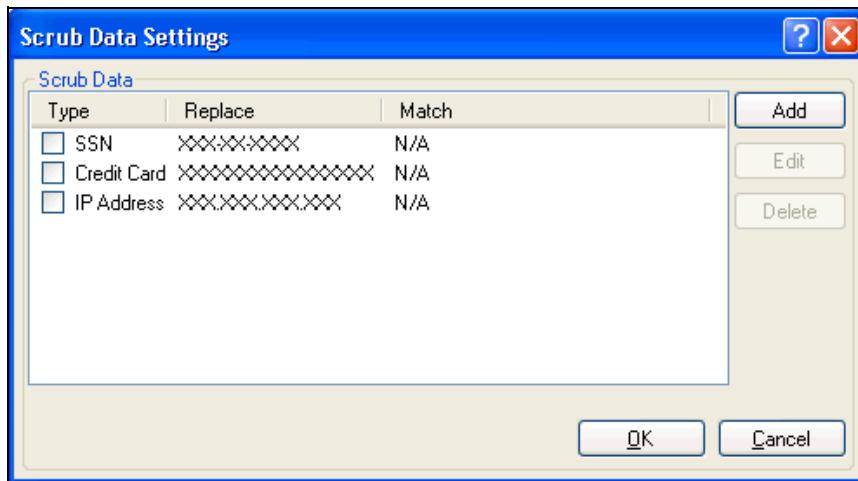
To send data to HP:

- 1 Click **Send to HP**.
The *Send to HP* dialog appears.
- 2 Select one or more installed products.

- 3 Choose which product-related items you want to send to HP Support.
- 4 If you select **Include an additional directory**, click **Browse** and identify the directory. The contents of that directory (and all subdirectories) will be uploaded to HP Support.
- 5 If you include scans, select the appropriate scan export options.
 - **Include scan logs:** Include log files associated with the selected scans.
 - **Only export scan logs (local scans only):** Do not include scan data.
 - **Scrub scan data:** Use a “scrubbing” feature that excludes sensitive data from the exported scan. To select specific scrubbing functions, click the **Configure** hyperlink to the right of the check box; see Scrubbing Data for instructions.
- 6 Click **Next**.
- 7 Enter the case number you obtained from HP support personnel (required).
- 8 To enter or modify customer information, click the **Customer Contact Information** hyperlink. First name, last name, and e-mail address are required.
- 9 Select an option from the Communication Settings group:
 - Send to HP Support via FTP
 - Send to HP Support via Secure Channel
 - Send the files to a local directory
- 10 Click **Send**.

Scrubbing Data

The Scrub Data Settings contain, by default, three non-editable regular expression functions that will substitute an X for each digit in a string formatted as a Social Security number, credit card number, or IP address.



To include these search-and-replace functions:

- 1 Select **Scrub Scan Data** in the **Scan Export Options** section.
- 2 Click **Configure**.
- 3 On the *Scrub Data Settings* window, select one or more of the functions in the **Type** column.

- 4 To create a Scrub Data function:
 - a Click **Add**.
 - b On the *Add Scrub Entry* window, select either **Regex** or **Literal** from the **Type** list.
 - c In the **Match** box, enter the string (or a regular expression representing a string) that you want to locate. If using a regular expression, you can click the browse button to open the Regular Expression Editor, with which you can create and test your regular expression.
 - d In the **Replace** box, enter the string that will replace the target specified by the **Match** string.
 - e Click **OK**.

Support Settings

Proxy

If you are not using a proxy server, select **Direct Connection** (proxy disabled).

If you are required to use a proxy server, select one of the following.

- **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 Internet Explorer proxy settings:** Import your proxy server information from Internet Explorer.
- **Use Firefox proxy settings:** Import your proxy server information from Firefox.

 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," or if the Internet Explorer setting "Use a proxy server for your LAN" is not selected, then a proxy will not be used. To access browser proxy settings:

Internet Explorer: **Tools** → **Internet Options** → **Connections** → **LAN Settings**

Firefox: **Tools** → **Options** → **Advanced** → **Network** → **Settings**

- **Configure a proxy using a PAC file:** Load proxy settings from the Proxy Automatic Configuration (PAC) file in the location you specify in the URL box.
- **Explicitly configure proxy:** Configure a proxy by entering the requested information.

SQL Server

To override SQL Server settings defined in WebInspect's application settings, select **Define local SQL Server settings** and supply the requested information. This feature is used most often to collect data from a different computer (that is, a machine other than the one on which this Support Channel software is running).

Server name

Enter or select the name of the server that will store WebInspect data.

Log on to the server

Specify the type of authentication used for the selected server:

- **Use Windows Authentication**—Log on by submitting the user's Windows account name and password.
- **Use SQL Server Authentication**—Use SQL Server authentication, which relies on the internal user list maintained by the SQL Server computer. Enter the user name and password.

Connect to a database

After supplying a server name, enter or select a specific database.

Advanced

Support Channel URL

If you are instructed to change the default Support Channel URL, do it here.

Communication Settings

Select one of the available protocols for sending files to the HP servers:

- **FTPS**—FTP Secure is an extension to the commonly used File Transfer Protocol (FTP) that adds support for the Transport Layer Security (TLS) and the Secure Sockets Layer (SSL) cryptographic protocols.
- **HTTPS**—Hypertext Transfer Protocol Secure is a combination of the Hypertext Transfer Protocol with the SSL/TLS protocol to provide encryption and secure identification of the server.

Log Level

Specify how different functions and events that occur within the Support Tool should be logged. The choices are (from most verbose to least verbose) Debug, Info, Warn, and Error.

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 WebInspect should submit when conducting a Web service scan.

Although the following procedure invokes the Web Service Test Designer from the WebInspect **Tools** menu, you can also open the designer from the HP Security Toolkit or through the WebInspect Scan Wizard by selecting **Start a Web Service Scan** from the WebInspect Start page and, when prompted, electing to launch the designer.



When the Web Service Test Designer is launched from the 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 WebInspect **Tools** menu or from the HP Security Toolkit.

- 1 Select **Tools** → **Web Service Test Designer**.
- 2 On the startup dialog, 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 and click **Import WSDL** .

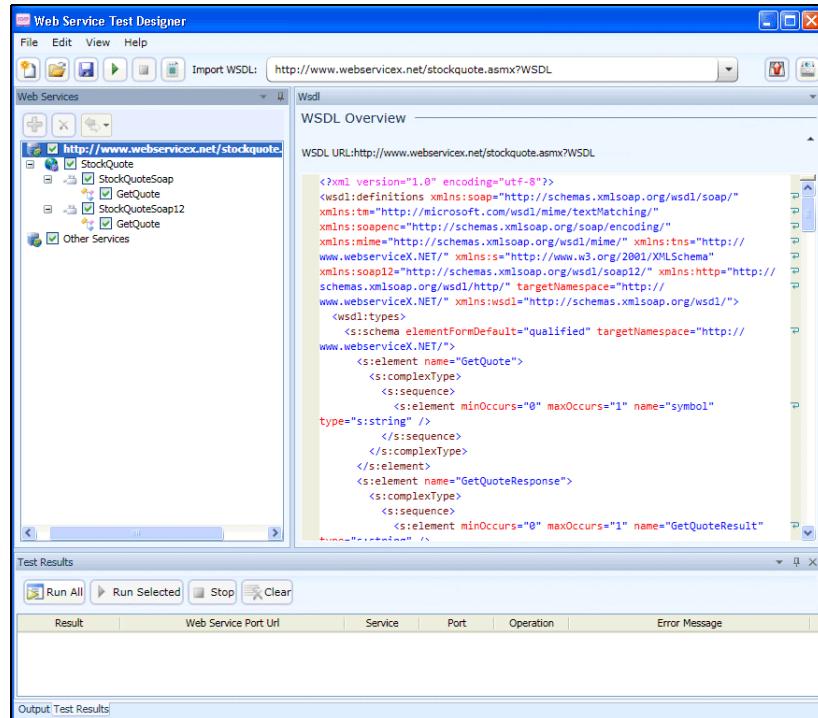
Example: `http://www.webservicex.net/stockquote.asmx?WSDL`.

- Click **Browse for WSDL**  and select a WSDL file that you previously saved locally.

If authentication is required, or if SOAP requests need to be made through a proxy server, see [Web Service Test Designer Settings](#) on page 435 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 430 for more information. Remove the check mark next to this item.

The WSDL endpoint (typically represented by a simple http URL string) appears in the left pane, followed by the service name and a hierarchical listing of the operations defined for that service. The right pane (by default) contains the WSDL URL and, when available, the namespace, binding namespace, and the port location.



The above illustration shows a simple WSDL that returns the current stock price and other related information when the user submits a corporate symbol used by the New York Stock Exchange.

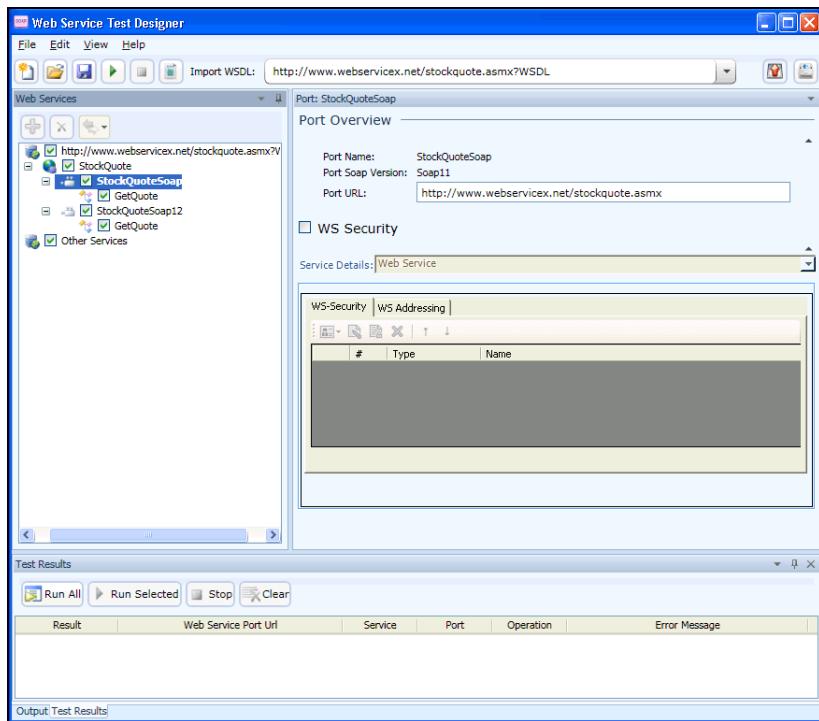
- 4 Select a service transport in the left pane to display the port information in the right pane. A port defines an individual endpoint by specifying an address for a binding. Note that if the description of the WSDL includes both SOAP version 1.1 and version 1.2, and if the operations in both descriptions 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.

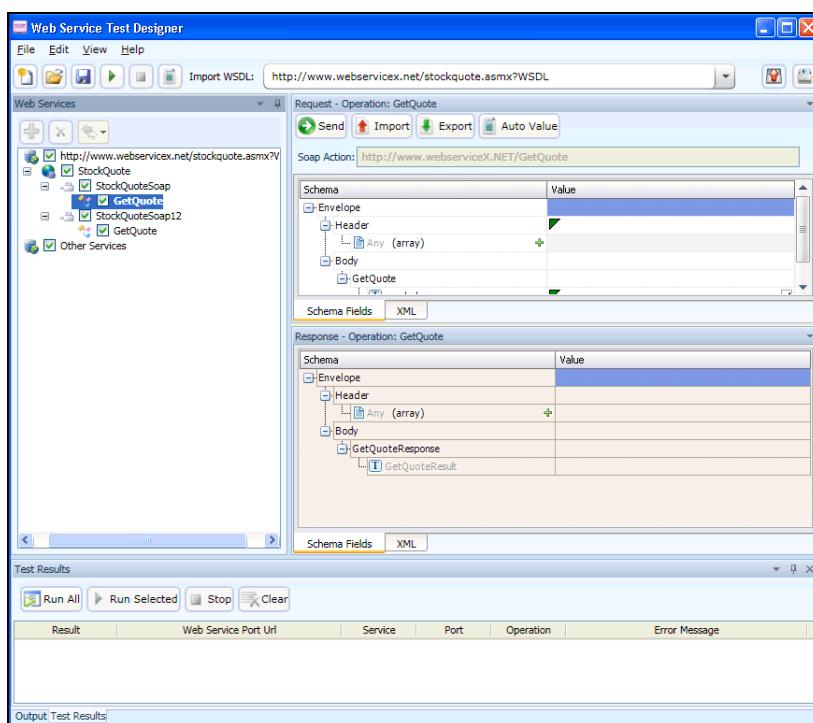
Port URL:	<input type="text" value="http://www.webservicex.net/stockquote.asmx"/>
<input type="checkbox"/> Include SOAP Action in HTTP Header	

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. This check box appears for SOAP 1.2 ports only and defaults to true.

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.



- 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 Settings](#) on page 423.
- 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).



- 7 Enter a value for each parameter in the operation. In this example, the user entered HPQ (the NYSE symbol for Hewlett Packard).

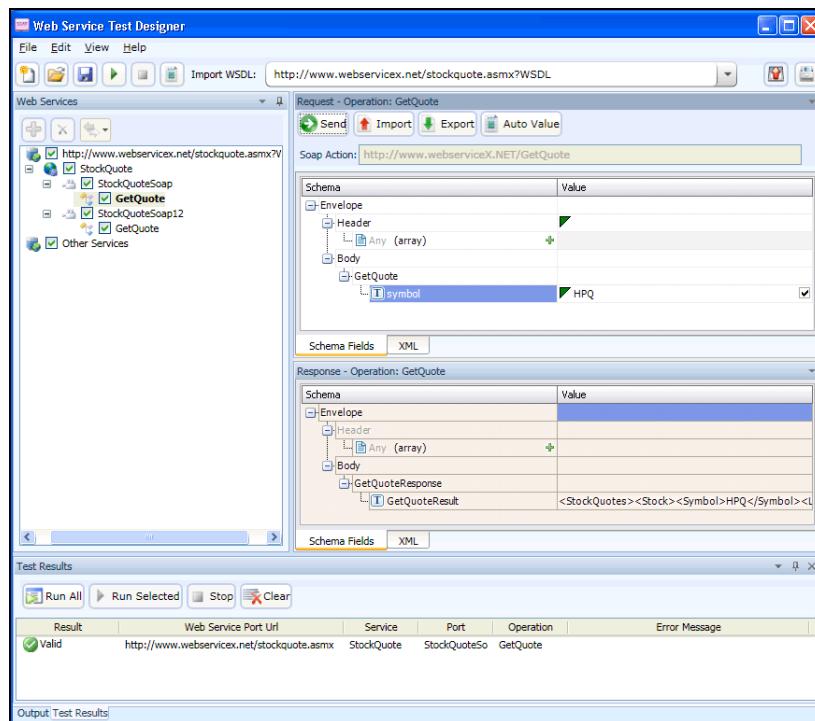
If you click **Auto Value**, the designer assigns a value to the operation. This value is either:

- 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 431 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 for more information.

- 8 Click **Send** .

Results appear in the lower portion. You can alternate between the Schema and XML views by clicking the appropriate tabs.



- 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, select a name and location for the Web Service Design file (.wsd).

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.

WS Security Settings

You can configure security settings for all operations in a Web service port, using a variety of services:

- Web Service
- Windows Communication Foundation (WCF) Service (CustomBinding)
- WCF Service (Federation)
- WCF Service (WSHttpBinding)

Select an appropriate service from the Service Details list and then provide the requested information.

Web Service

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

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.

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.

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.

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)

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. See [Advanced Security Settings](#) on page 428.

WCF Service (Federation)

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. See [Advanced Security Settings](#) on page 428.

WCF Service (WSHttpBinding)

Using WCF Service (WSHttpBinding), you can choose from several types of authentication: None, Windows, Certificate, or Username (message protection).

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 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.

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)

Impersonation Level	Description
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.

Enable secure session. Allows a secure session using Certificate type authentication.

Username (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

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

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.

Drop down section Security Tab

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

This tab lets you set the HTTP and Proxy information for your test.

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.

Manually Adding Services

You may encounter a Web service that does not have a WSDL associated with it.

For example, the 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 probably will 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.

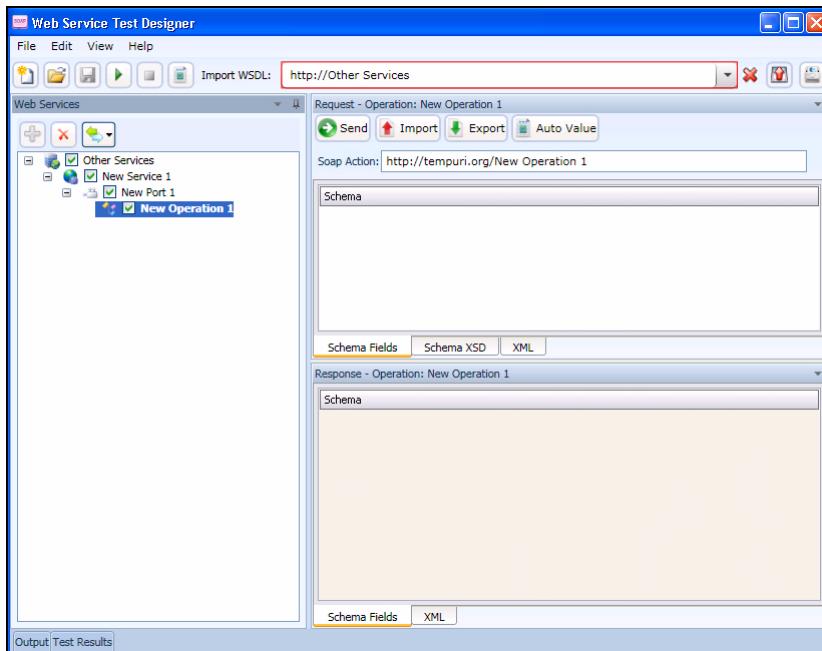
- 2 If authentication is required, select **WS Security** and provide the required credentials.

- 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**.



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.

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>HPQ</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.

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**.

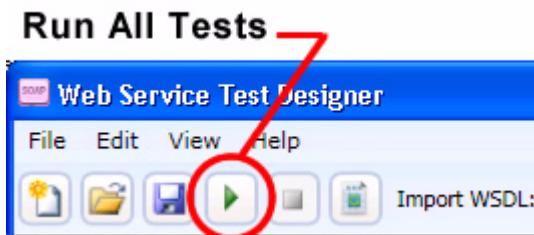
The Web Service Test Designer submits the service request, with values inserted for each operation.

- 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.

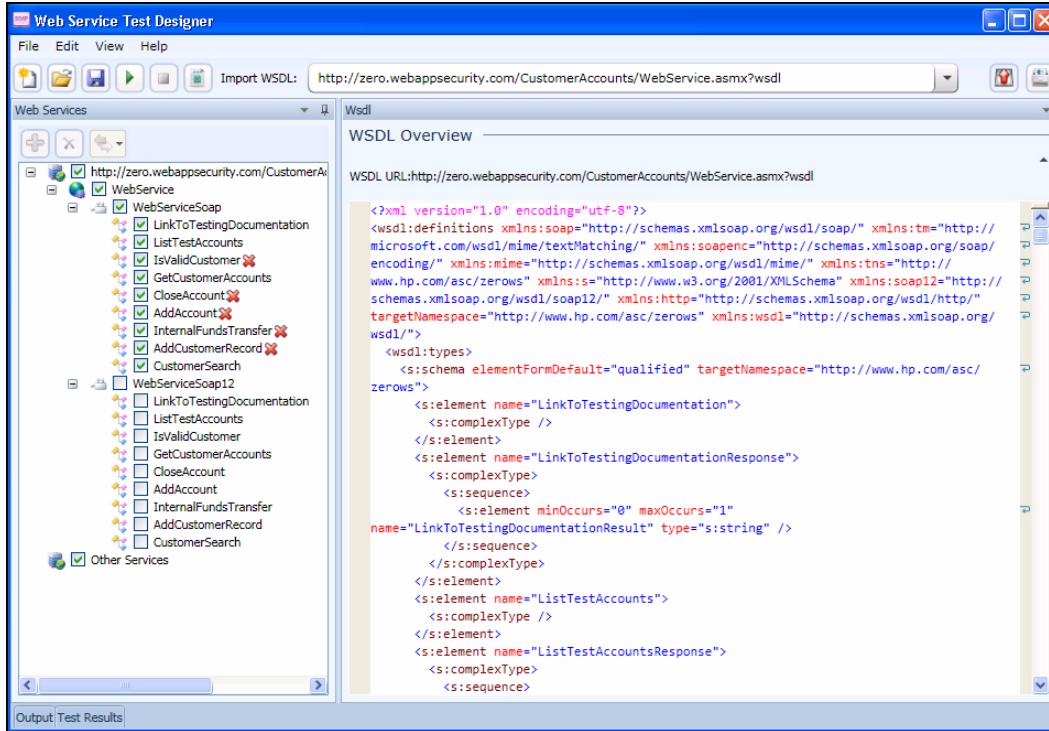
Testing Your Design

You can, at any time, test the configuration of any or all operations.

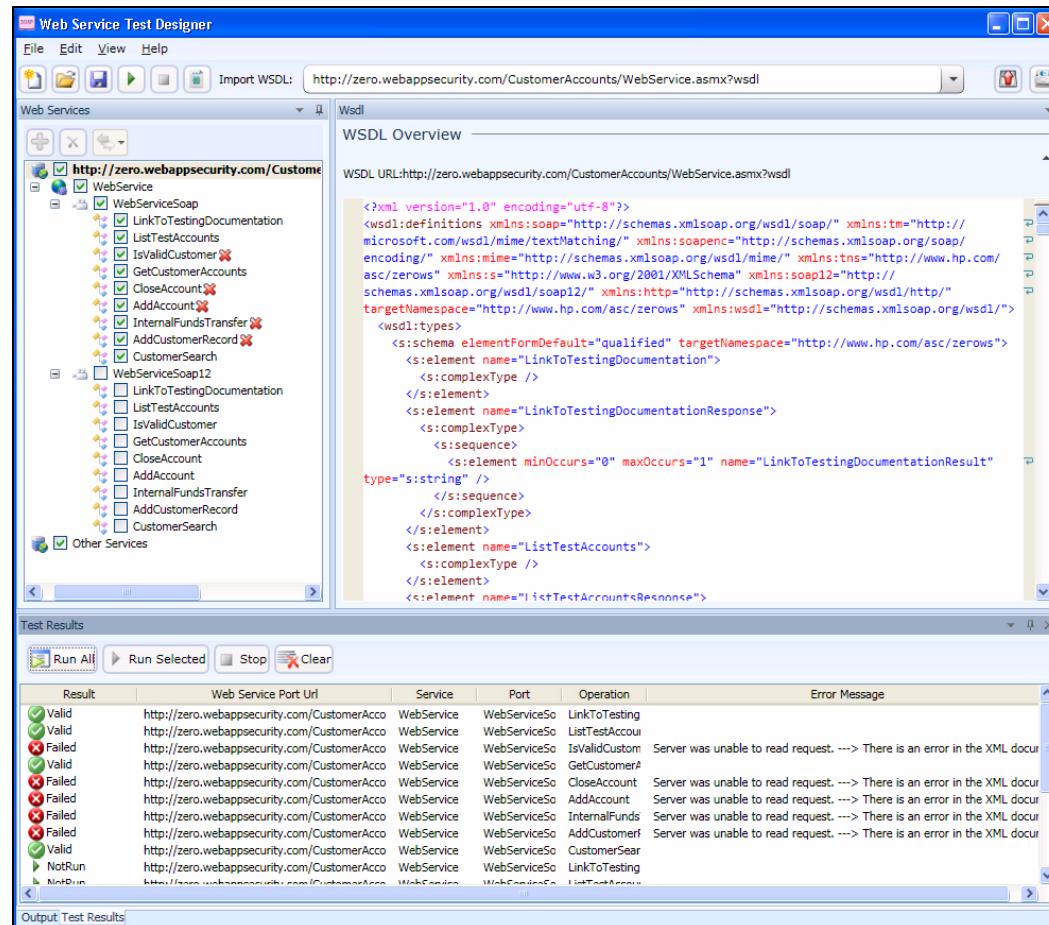
After importing the WSDL, click **Run All Tests**.



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.



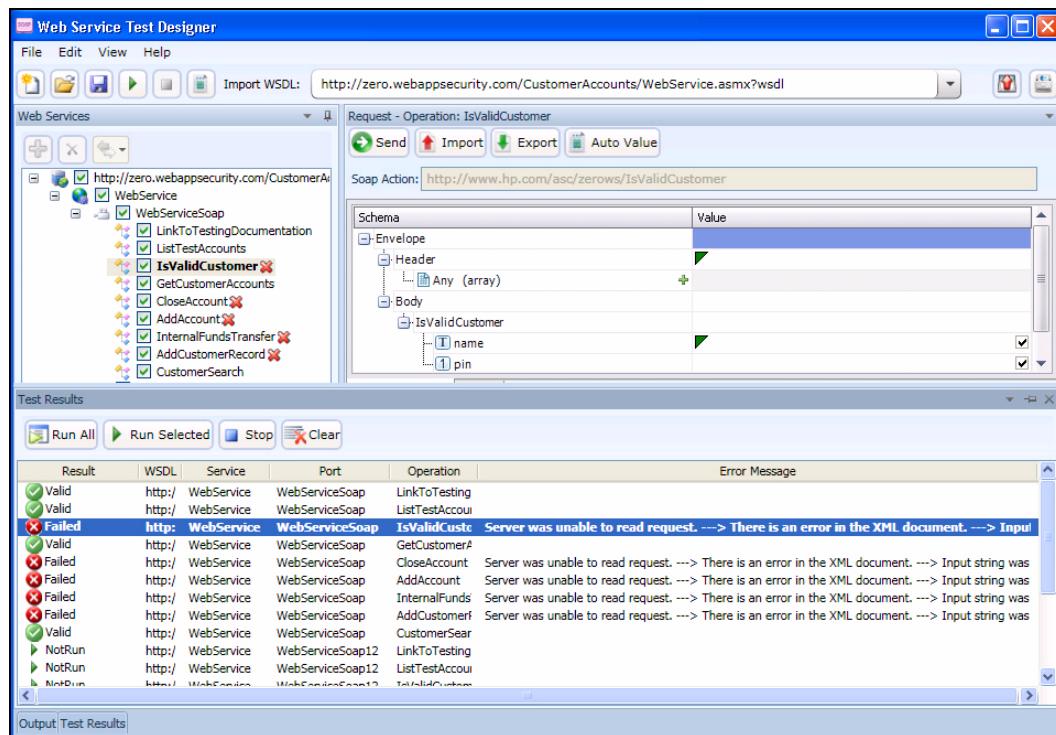
The Test Results pane displays the following information:

- 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.
- WSDL – The WSDL 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.



Web Service Test Designer Settings

The Web Services Designer has two categories of settings:

- Network Proxy
- Network Authentication

To access settings, click **Edit** → **Settings**.

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 Internet Explorer:** Import your proxy server information from Internet Explorer.
 - **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.
- 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:
 - 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** box. Use commas to separate entries.
- 4 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.

Web Application Firewall Integration Tool

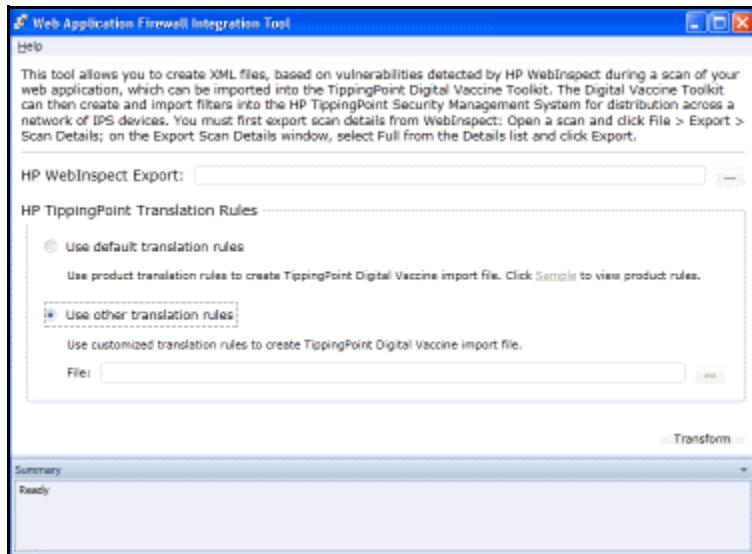
The Web Application Firewall Integration Tool allows you to create a file that can be imported into the TippingPoint Digital Vaccine Toolkit, based on vulnerabilities detected by HP WebInspect during a scan of your web application. The Digital Vaccine Toolkit can then create and import filters into the HP TippingPoint Security Management System for distribution across a network of Intrusion Prevention System (IPS) devices.

Task 1: Export scan details from WebInspect

- 1 Open a scan and click **File → Export → Scan Details**.
- 2 On the *Export Scan Details* window, select **Full** from the **Details** list.
- 3 Click **Export**.
- 4 On the *Save As* dialog, specify a name and location for the file and click **Save**.

Task 2: Create an XML file from the Scan Details

- 1 Click **Start → All Programs → HP → HP Security Toolkit → Web Application Firewall Integration**.



- 2 In the **WebInspect Export** box, enter the location and name of the scan details file you created in Task 1.

Click the browse button to invoke a standard file-selection dialog.

- 3 Under **HP Tipping Point Translation Rules**, select one of the following:
 - **Use default** - Use the standard translation rules.
 - **Use other** - Use a custom set of rules developed by you or a third party. If you select this option, click the browse button to identify the file containing the translation rules.
- 4 Click **Transform**.
- 5 Using a standard file-selection dialog, specify a location for the XML files, and then click **Save**.

The following files will be created in the specified location for import into the Tipping Point Digital Vaccine Kit:

- TpSqlnjRules.xml
- TpFileIncludeRules.xml
- TpXssRules.xml

A Attacks and Methodologies

Introduction

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

WebInspect is a standalone software package that 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 such as the Web, having a security tool that's always up to date is an absolute necessity. With this in mind, WebInspect'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 HP's security experts, as well as a wide variety of leading third-party security organizations and analysts.

When new methods of attack are discovered, WebInspect is ready with same-day upgrades to its SecureBase vulnerabilities database. WebInspect employs the following list of attacks and key methodologies when assessing the security vulnerabilities of your Web application. This list does not include all attacks and methodologies employed by WebInspect.

- Parameter Manipulation
- Path Manipulation
- Web Server Assessment
- Site Searching
- Application Mapping
- Brute Force Authentication Attacks
- Content Investigation
- Known Attacks

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. These attacks are directed specifically toward query strings, postdata, headers, and cookies.

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, WebInspect implements advanced query string manipulation to determine the application's vulnerability to query string manipulation.

Postdata

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. WebInspect will determine your application's susceptibility to attacks that rely on the POST method of parameter manipulation.

Headers

Both HTTP requests and responses utilize 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 "referer" or "user-agent" to a database for traffic statistics. WebInspect will intercept header information, and attempt 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. WebInspect will list all cookies discovered during a scan, and attempt to change their parameters during an audit.

In general, parameter manipulation attacks belong to one of the following four categories:

- Injection
- Overflow

- Addition
- Deletion

Parameter Injection

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

Example:

The string

`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:

In response to the following request

`http://www.site.com/article.pl?id=;id;`

a server might send data such as

`uid=99(nobody) gid=99(nobody).`

SQL injection

SQL injection attack strings are composed of fragments of SQL syntax that are 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)+'`

Directory traversal-Directory traversal attack strings are expressions that will cause the Web application to display the contents of files above the webroot if the Web application uses the string to specify a file location without first completely parsing out traversal characters.

Example:

In response to the following request

`http://www.site.com/article.asp?id=../../../../boot.ini`

a server might send data similar to

`:\\boot.ini file contents`

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 that 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:

- Search engines that repeat back the search keyword that was entered
- Error messages that repeat back 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.

Example:

```
'<script>window.open('http://www.website.com')</script>
```

Abnormal input

Abnormal input attack strings are composed of characters that can cause unhandled exceptions 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
```

Hidden content access

Hidden content access strings are names or numbers that will cause a Web application to display hidden content, such as an administrative interface or unused portions of the website.

Example:

Given the query

`http://www.site.com/index.php?ArticleID=2`

if valid values for ArticleID range from 1-30, then changing the query to

`http://www.site.com/index.php?ArticleID=99`

might reveal a hidden portion of the website not normally accessed by users.

Untrusted application access

Untrusted application access will change existing values that are Boolean or single-digit numbers to other values in an attempt to gain access to an authenticated portion of the website.

Example:

Change this query

`http://www.site.com/index.php?LoggedIn=false`

to

`http://www.site.com/index.php?LoggedIn=true`

Format string attack

A format string attack will test the application for improper format handling on user input.

Example:

Changing this query

`http://www.site.com/search.cgi?query=contact`

to

`http://www.site.com/search.cgi?query=%x%x%x%`

will break the application and allow the user to gain control of the webserver.

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.

Numeric overflow

A numeric overflow increases an existing digit value to extremely high values in order to invoke an application error.

Example:

Changing this query

`http://www.site.com/index.php?ArticleID=2`

to

`http://www.site.com/index.php?ArticleID=2147483648`

will attempt to exceed the value of a signed integer.

String overflow

A string overflow increases the length of an alphabetic parameter in order to invoke an application error.

Example:

Changing this query

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

by adding several thousand characters (as below)

`http://www.site.com/webapp.asp?XXXXXXXX...XXX=ValidArgument`

will attempt to exceed the value of a signed integer.

Parameter Addition

Parameter addition attacks insert new parameters into an HTTP request. Parameter addition attacks can be used to gain access to restricted or undocumented application features, manipulate internal application settings.

Application debug/backdoor modes

These parameters are often undocumented application features that are added by programmers in order to assist with quality assurance. Debug and backdoor modes access 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`

Internal parameter specification

These attacks define parameters and arguments in the HTTP request that are only supposed to be set inside of the Web application. If the Web application sets the variable specified in the parameter to the value specified in the argument, internal application settings will be altered.

Example:

`http://www.site.com/`

`webapp.php?ValidParm=ValidArgument&DBUsername=sa`

Parameter Deletion

Parameter deletion removes a parameter value from query data or cookie headers. If the Web application relies on the presence of the removed parameters for basic functionality, and unhandled exception may occur. 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.

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 webroot, bypass authorization settings, display directory listings or display file source.

Path truncation

Path truncation attacks are requests for known directories without filenames. This may cause directory listings to be displayed.

Example:

Given the following URL

`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/`

may cause the webserver to reveal directory contents or to cause unhandled exceptions.

Case sensitivity

Case sensitivity attacks change the case of the characters in the filename in an attempt to change the manner in which the request is processed. If the Web application performs a case-sensitive string comparison for authorization or processing purposes this may defeat authorization settings or cause source code to be disclosed.

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 parsing the encoded characters first, authorization settings maybe defeated or source code may be disclosed.

- **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. WebInspect submits strings that have been converted to their Unicode equivalent, and attempts to gain unauthorized authentication credentials through this manipulation.
- **Hex-encode:** Hex-encoding involves replacing strings with their hexadecimal equivalent. WebInspect submits hexadecimal-encoded strings, and attempts to gain unauthorized authentication credentials through this manipulation.

MS-DOS 8.3 Short Filename

These attacks convert the filenames to the MS-DOS 8.3 format. 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:

The file named
longfilename.asp
would become
longfi~1.asp

Directory traversal

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

Example:

../../../../boot.ini

Character stripping

These 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 attacks add a special character to the end of a file or directory name. For example, file.asp would become file.asp%00.
- Character prepend attacks add a special character to the beginning a file or directory name. For example, /protecteddirectory/file.asp would become %00protecteddirectory/file.asp.
- Buffer truncation attacks add a large number of characters to a filename and extension followed by another extension. If a Web application determines how the request should be processed based on the file extension, it will protect against a buffer overflow by truncating the request at a certain length and then stripping out the special characters, in that order. Authorization settings may be bypassed and source code may be displayed using the following example:

Webserver Assessment

During a Web server assessment, WebInspect tests 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. WebInspect will ascertain 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

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

HTTP Method Support

WebInspect determines what HTTP methods are supported by the Web server.

Site Searching

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 and application source code. WebInspect determines the availability of the following items, among others, to users of your Web application.

Test files

Test files 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

Administrative interfaces are applications that network administrators often place on a network to conduct remote maintenance.

Program dumps

When a program terminates prematurely, it often leaves a dump file on the server (an image of system memory when the program stops executing). 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

A Site Statistics page 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

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 WebInspect 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

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

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. 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 Web site. 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. 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. 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.

Brute Force Authentication Attacks

This test determines if users are employing usernames and passwords that an unauthorized intruder might be able to guess easily. For example, it will discover if an authorized user is accessing a Web site by entering a username of “customer” and a password of “password”; the Web administrator could then warn that user about the susceptibility and suggest changing the password and/or username.

Web Brute will attempt a “brute force” attack of three authentication types: HTTP Basic, NTLM, and forms on Web pages.

Content Investigation

Content Investigation involves searching through content discovered during a site search to determine what information available to users of your Web application should remain private. 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 a site.
- **WebServer/Web Package Identification:** 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 header evidence, link evidence, and default/template page evidence.
- **Absolute Path Detection:** WebInspect detects if 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.
- **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:** 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.
- **Session Hijacking:** WebInspect sends multiple requests in order to gain multiple sessions and analyze the session ID within the URLs or cookies for weaknesses.
- **Non-Restrictive Search Engine:** Determining if a search engine's search scope is unrestricted. Many search engines have access to search the entire webroot or the entire drive of a Web server. If this is available, a search for "admin" or "adduser" will return links to the administrative portion of the Web site.

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. WebInspect relies on a proprietary database, which contains fingerprints of known attacks dating back to the birth of the World Wide Web. WebInspect downloads checks for new risks and exploits each time customers run Smart Update, ensuring that the product is always at the forefront of hacking expertise.

B Policies and Components

Introduction

A policy is a collection of vulnerability checks and attack methodologies that WebInspect deploys against a Web application. Each policy is kept current through WebInspect's Smart Update functionality, ensuring that scans are accurate and capable of detecting the most recently discovered threats.

Policies

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

This group contains a policy designed to test applications for the most pervasive and problematic Web application security vulnerabilities as determined by HP.

- **Standard:** 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.
- **OWASP Top 10 Application Security Risks 2010:** 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.

Hazardous

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

- **Assault:** 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 be used only in test environments.

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



Warning: An All Checks scan includes checks that may write data to databases, submit forms, and create denial-of-service conditions. It is strongly recommended that All Checks scans be used only in test environments.

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.
- **Application:** 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.
- **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.
- **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 Web site 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.
- **Dev:** 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 Developer policy does not execute checks that are likely to create denial-of-service conditions, so it is safe to run on production systems.
- **OWASP Top 10 Application Security Risks 2007:** 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.

- **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.
- **Platform:** 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:** 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:** 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:** 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.
- **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.

Custom

The Custom group contains all user-created policies, any custom policies modified by a user, and the policy listed below.

Hacme Bank: A custom policy for scanning an example Web site maintained by OWASP. For more information, visit https://www.owasp.org/index.php/Hacme_Bank.

Policy Components

Policy components are organized into the following groups:

- | | |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none"> • Audit Engines • General Application Testing • General Text Searching • Third-Party Web Applications | <ul style="list-style-type: none"> • Web Frameworks/Languages • Web Servers • Web Site Discovery • Custom Checks |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|

For detailed information about all the possible agents, open the Policy Manager, select the Attack Groups category, and click on any agent name.

Audit Engines

WebInspect uses the following audit engines.

Audit Options: These include Robots.txt Parser, WebInspect Scan Signature, Ws_ftp.log Parser, and CVS Entries Parser.

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, WebInspect continuously applies appropriate scan resources that adapt to the specification application environment.

- **Comment Checks:** The comment audit examines each session for filenames 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 runs the cross-site scripting parameter injections attacks. Cross-site scripting is caused by insufficient filtering of client-supplied data that is returned to Web users by the Web application.
- **Directory Enumeration:** Directory enumeration finds all directory paths and possibilities on the application server, including hidden directories that could possibly contain sensitive information. This helps WebInspect create a full and accurate map of the targeted site.
- **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. Data extension checking involves adding file extensions to find old renamed files left on the server. For example, an attacker might find hi.asp, and then search for hi.asp.bak or hi.asp.old. WebInspect will attempt to locate all files left on your server that could be used by an attacker.
- **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. For example, an attacker who finds hi.asp might search for *copy of hi.asp* and retrieve the script's source code.
- **Fixed Checks:** This audit performs checks for files with known vulnerabilities. The Fixed Checks audit does not probe the directory structure before sending the attacks.
- **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 Web site 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 Web site.
- **Known Vulnerabilities:** This audit engine examines your Web site 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.
- **Logic Checks:** This audit performs checks based on previously discovered vulnerabilities.
- **Postdata Injection, 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. WebInspect will determine your application's susceptibility to attacks that rely on the POST method of parameter manipulation.
- **Query Injection, 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, 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.

- **Request Inspect:** During the crawl of a Web application to map its internal structure, the Request Inspect engine applies the regular expressions that are associated with checks to the requests being sent.
- **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.
- **Server Side Include:** During the course of normal operations, many Web applications will accept a full URL as an expected and returned parameter value. This audit engine will manipulate that process and determine if an attacker could exploit any vulnerabilities within the application by including commands and other functions within the URL accepted by the application.
- **Site Search:** This can be considered the information-gathering stage, employing the same tactics an intruder would use 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 reveal confidential data, information about internal server and application configurations and settings, administrative access to the site, and application source code.

- **SOAP Scan:** 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 scan 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

General Application Testing

This group of agents, used mainly by the Directory Enumeration engine, searches the site's tree structure for commonly occurring directories. 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).

General Text Searching

This group of agents searches for a wide variety of text strings, such as database connection strings, error messages, Social Security numbers, credit card numbers, and debug applications.

Third-Party Web Applications

This group of agents looks for known vulnerabilities associated with hundreds of 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 specific Web servers.

Web Site Discovery

This group of agents searches for commonly used files, account information, backup files, CVS files, Include files, core dumps, statistics, logs, and various other files that could be used to infiltrate and exploit the Web site.

Custom Checks

A custom check is a user-defined probe for a specific vulnerability that the standard WebInspect repertoire does not address. Use the Policy Manager to create custom checks and integrate them into your policies. See [Creating a Custom Check](#) on page 227 for more information.

C HTTP Status Codes

Introduction

The following list of status codes was extracted from the Hypertext Transfer Protocol version 1.1 standard (rfc 2616) developed by the Internet Society. You can view the complete standard at <http://www.w3.org/Protocols/rfc2616/rfc2616.html>.

Table 2 HTTP Status Codes

Code	Definition
100	Continue
101	Switching Protocols
200 OK	Request has succeeded
201 Created	Request fulfilled and new resource being created
202 Accepted	Request accepted for processing, but processing not completed.
203 Non-Authoritative Information	The returned metainformation in the entity-header is not the definitive set as available from the origin server, but is gathered from a local or a third-party copy.
204 No Content	The server has fulfilled the request but does not need to return an entity-body, and might want to return updated metainformation.
205 Reset Content	The server has fulfilled the request and the user agent should reset the document view which caused the request to be sent.
206 Partial Content	The server has fulfilled the partial GET request for the resource.
300 Multiple Choices	The requested resource corresponds to any one of a set of representations, each with its own specific location, and agent-driven negotiation information (section 12) is being provided so that the user (or user agent) can select a preferred representation and redirect its request to that location.
301 Moved Permanently	The requested resource has been assigned a new permanent URI and any future references to this resource should use one of the returned URIs.
302 Found	The requested resource resides temporarily under a different URI.
303 See Other	The response to the request can be found under a different URI and should be retrieved using a GET method on that resource.
304 Not Modified	If the client has performed a conditional GET request and access is allowed, but the document has not been modified, the server should respond with this status code.

Table 2 HTTP Status Codes (cont'd)

Code	Definition
305 Use Proxy	The requested resource MUST be accessed through the proxy given by the Location field.
306 Unused	Unused.
307 Temporary Redirect	The requested resource resides temporarily under a different URI.
400 Bad Request	The request could not be understood by the server due to malformed syntax.
401 Unauthorized	The request requires user authentication. The response MUST include a WWW-Authenticate header field containing a challenge applicable to the requested resource.
402 Payment Required	This code is reserved for future use.
403 Forbidden	The server understood the request, but is refusing to fulfill it.
404 Not Found	The server has not found anything matching the Request-URI.
405 Method Not Allowed	The method specified in the Request-Line is not allowed for the resource identified by the Request-URI.
406 Not Acceptable	The resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.
407 Proxy Authentication Required	This code is similar to 401 (Unauthorized), but indicates that the client must first authenticate itself with the proxy.
408 Request Timeout	The client did not produce a request within the time that the server was prepared to wait.
409 Conflict	The request could not be completed due to a conflict with the current state of the resource.
410 Gone	The requested resource is no longer available at the server and no forwarding address is known.
411 Length Required	The server refuses to accept the request without a defined Content-Length.
412 Precondition Failed	The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.
413 Request Entity Too Large	The server is refusing to process a request because the request entity is larger than the server is willing or able to process.
414 Request-URI Too Long	The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.
415 Unsupported Media Type	The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

Table 2 HTTP Status Codes (cont'd)

Code	Definition
416 Requested Range Not Satisfiable	A server should return a response with this status code if a request included a Range request-header field, and none of the range-specifier values in this field overlap the current extent of the selected resource, and the request did not include an If-Range request-header field.
417 Expectation Failed	The expectation given in an Expect request-header field could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.
500 Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request.
501 Not Implemented	The server does not support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.
502 Bad Gateway	The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.
503 Service Unavailable	The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.
504 Gateway Timeout	The server, while acting as a gateway or proxy, did not receive a timely response from the upstream server specified by the URI (e.g., HTTP, FTP, LDAP) or some other auxiliary server (e.g., DNS) it needed to access in attempting to complete the request.
505 HTTP Version Not Supported	The server does not support, or refuses to support, the HTTP protocol version that was used in the request message.

Glossary

Audit

To assess your Web application for security vulnerabilities.

Authentication

Identity verification, typically through the use of logon passwords. Authentication precedes authorization. WebInspect supports HTTP Basic, NTLM, Automatic, Kerberos, Digest, Integrated Windows, and Web-based (form) authentication.

Authorization

Access control. After a user has been authenticated (proven their identity, typically via a logon password), the operating system or application must identify what resources the user can access during this session, and authorize access accordingly.

AVDL

Application Vulnerability Description Language. An interoperability standard developed by leading application security vendors and approved by the Organization for the Advancement of Structured Information Standards (OASIS) AVDL Technical Committee as a Committee Draft. The goal of AVDL is to create a uniform method of describing application security vulnerabilities using XML.

Banner

Server identification. An attacker will grab banners to determine the make and model of the server operating system, and use that information when formulating an attack against the vulnerabilities of that software package.

Canonicalization

Sanitizing data by not accepting improper input. For example, stripping special characters from a request before processing it.

Certificate

A certificate states that a specific Web site is secure and genuine. It ensures that no other Web site can assume the identity of the original secure site. When sending personal information over the Internet, users should check the certificate of the Web site to ensure that it will protect personally identifiable information. When downloading software from a Web site, certificates verify that the software is coming from a known, reliable source. A security associate associates an identity with a public key. Only the owner of the certificate knows the corresponding private key, which allows the owner to make a "digital signature" or decrypt information encrypted with the corresponding public key.

Client

A client is the requesting program or user in a client/server relationship. For example, the user of a Web browser is effectively making client requests for pages from servers all over the Web. The browser itself is a client in its relationship with the computer that is getting and returning the requested HTML file. The computer handling the request and sending back the HTML file is a server.

Common Weakness Enumeration (CWE)

Common Weakness Enumeration is a software community project that aims at creating a catalog of software weaknesses and vulnerabilities. The goal of the project is to better understand flaws in software and to create automated tools that can be used to identify, fix, and prevent those flaws. The project is sponsored by Mitre Corporation.

Crawl

The process by which WebInspect identifies the structure of the target Web site. This is usually followed by an audit, which is the actual vulnerability scan. A crawl and an audit, when combined into one function, is termed a scan.

Cookie

Cookies are information stored by a server on a client for future use (such as user preferences, configuration information, etc.). Cookies appear in two basic forms, as individual files or as records within one contiguous file. Often, there are multiple sets, the result of multiple browsers being installed in differing locations. In many cases, it is the forgotten cookies that contain the revealing information that you would prefer others not see.

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, allowing the attacker to execute script on the machine of any user that views the malicious page. Any site that allows users to post text messages can be vulnerable to an attack such as this.

Custom Check

A custom check, like a custom attack agent, is a user-defined probe for a specific vulnerability that the standard WebInspect repertoire does not address. The major difference is that a custom check can be created using a simple wizard, while a custom attack agent is programmed in Visual Basic using a special integrated development environment and requires significant knowledge of the WebInspect architecture. The result is also more narrow in scope than a typical attack agent.

Digest Authentication

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.

HIPAA

Health Insurance Portability and Accountability Act .

HTTP

Hyper Text Transfer Protocol. HTTP is the set of conventions that governs how HTML documents are transmitted and received across the World Wide Web. When browsing Web sites, your Web browser is a client program that makes requests (for example, that a certain Web page be displayed) from a Web server somewhere on the Internet. An important element of HTTP is in how servers (the computers hosting the Web applications, in this instance) handle requests from clients (remote computers connecting to the server via the World Wide Web). A session can be defined as the matched pair of a client request and a server response. HTTP is a stateless protocol-no concept of session state is maintained by HTTP when handling client-server communications. While that sounds complicated, it is really quite simple when broken down. Each request made by a client is handled individually by a server. Multiple requests made by the same client are each treated as unique by the responding server. In other words, the server does not attempt to maintain a connection with the client at any time.

HTTP Basic Authentication

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 a user's credentials are rejected, the browser displays an authentication window to re-enter the user's credentials. Internet Explorer allows the user three connection attempts before failing the connection and reporting an error to the user. If the Web server verifies that the user name and password correspond to a valid user account, a connection is established.

The advantage of HTTP 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.

IDE

Integrated Development Environment; a programming environment integrated into an application.

IDS

Intrusion Detection System. This type of system supplements perimeter security applications (such as a firewall) and identifies attacks that have passed through those defenses.

Image map

In Internet development, an image map is a graphic defined so that different areas of the image are linked to different destinations.

Kerberos Authentication

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

Login Macro

This type of macro is used for Web form authentication. You can also incorporate logic that will prevent WebInspect from terminating prematurely if it inadvertently logs out of your application.

MIME Type

Multipurpose Internet Mail Extensions (MIME) is a specification for formatting non-ASCII messages so they can be sent over the Internet. The Content-Type header indicates the type and subtype of the message content, for example

Content-Type: text/plain

The combination of type and subtype is generally called a **MIME type** (also known as Internet media type). Examples include:

- text/html
- image/jpeg
- image/gif
- audio/x-wave
- audio/mpeg
- video/mpeg
- application/zip

Navigation Pane

When conducting or viewing a scan, the navigation pane is on the left side of the WebInspect window. It includes the Site, Sequence, Search, and Step Mode buttons, which determine the contents (or "view") presented in the navigation pane.

NTLM Authentication

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.

Parameter

An item of information, such as a name, a selection, or a number, passed to a program by another program or an end-user.

Policy

The set of vulnerability checks and attack methodologies that WebInspect will deploy against a Web application.

Proxy server

In Internet terminology, a proxy server is one that serves as an intermediary between a workstation user and the actual Internet. Requests for Internet services made by the client (the workstation) must pass through the proxy server, as also do the Web server responses. A

proxy server can be used to ensure network security, provide adequate caching purposes, and regulate administrative control.

Query string

The extra bit of data in the URI after the question mark that is used to pass variables. The query string is used to transfer data between the client and the server. 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.

Scan

A generic term for the investigation of a Web site or enterprise. The actual task may be either a crawl, audit, or a combined crawl and audit.

Sensor

A sensor is the WebInspect application when connected to AMP for the purpose of performing remotely scheduled or requested scans with no direct user interaction through its graphical user interface. It receives its instructions exclusively from the configurable connection to an AMP Manager.

Server

In the Web application client/server model, a server (a program housed in a computer) uses HTTP to serve files that form Web sites to users. The user's system contain an HTTP client (e.g. the Web browser) that forwards requests to the Web server, which responds with the appropriate data. Two leading Web servers are Apache and Microsoft's Internet Information Server (IIS).

Session

A session is a matched set containing both the client request and server response. For Internet applications, each session is associated with a particular port. A session may contain up to three components:

- A link to a URL
- The HTTP request that WebInspect generates as a result of that link (but does not necessarily send)
- The associated HTTP response returned by the server

Session hijacking

Allows an attacker to masquerade as another user and gain access to Web service without having to authenticate. By using session hijacking, an attacker has access to the Web application with permissions of the original user.

Session ID

Generally, successful authentication credentials stored so that a user does not have to enter them repeatedly. Since the session ID can be used instead of a user name and password combination, an attacker who discovers and provides a valid session ID in a request could perform session hijacking or replay attacks.

Seven Pernicious Kingdoms

Seven Pernicious Kingdoms (7PK) is a taxonomy of software security errors developed by the Fortify Software Security Research Group together with Dr. Gary McGraw. Each vulnerability category is accompanied by a detailed description of the issue with references to original sources and code excerpts, where applicable, to better illustrate the problem.

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

The seven kingdoms are:

1. Input Validation and Representation
2. API Abuse
3. Security Features
4. Time and State
5. Errors
6. Code Quality
7. Encapsulation

The primary goal of defining this taxonomy is to organize sets of security rules that can be used to help software developers understand the kinds of errors that have an impact on security. By better understanding how systems fail, developers will better analyze the systems they create, more readily identify and address security problems when they see them, and generally avoid repeating the same mistakes in the future. For more information, see <http://www.hpenterprisesecurity.com/vulncat/en/vulncat/index.html>.

Smart Scan

Smart Scan is an "intelligent" feature that discovers the type of server that is hosting the Web site and checks for known vulnerabilities against that specific server type. For example, if you are scanning a site hosted on an IIS server, WebInspect will probe only for those vulnerabilities to which IIS is susceptible. It would not check for vulnerabilities that affect other servers, such as Apache or iPlanet.

Smart Update

This WebInspect feature contacts the Hewlett-Packard data center via the Internet to check for new or updated adaptive agents, vulnerability checks, and policy information. Smart Update will also ensure that you are using the latest version of WebInspect, and will prompt you if a newer version of the product is available for download.

SOAP

Simple Object Access Protocol. SOAP uses HTTP and XML as the means to exchange information so that programs on one platform (for example, Windows XP) can communicate with a program on the same or a different operating system (such as Linux).

SQL injection

The act of passing SQL code not intended by the developer into an application. For example, 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.

Stack Trace

This feature is designed to support HP Fortify SecurityScope when it is installed and running on the target server. For certain checks (such as SQL injection, command execution, and cross-site scripting), SecurityScope intercepts WebInspect HTTP requests and conducts runtime analysis on the target module. If this analysis confirms that a vulnerability exists, SecurityScope appends the stack trace to the HTTP response. Developers can analyze this stack trace to investigate areas that require remediation.

Startup Macro

This type of macro is used most often to focus on a particular subsection of the application. It specifies URLs that WebInspect will use to navigate to that area. It may also include login information, but does not contain logic that will prevent WebInspect from logging out of your application.

Step mode

Step mode captures each click followed on the site and then develops the site structure. Once you have completed clicking through the site, click **Audit** to assess the security vulnerabilities of the site.

String

A block of values or symbols, such as a character string (a sequence of alphanumeric characters), or a binary string (a sequence of binary values).

Trojan

A Trojan horse attack is the programming equivalent of the wooden horse given to the city of Troy. Seemingly benign data or programming is used to hide malicious or harmful code in such a way that it can instigate its chosen form of damage without your knowledge.

URI

Uniform Resource Identifier. According to the World Wide Web Consortium, Internet space is populated by many points of content. URIs are the method used to locate any given point of content on the Internet, whether it be a Web page, a video or music file, a program, or a graphic image. A URL (Uniform Resource Locator) is a particular form of URI, and is used as a designation for a Web page address. Typically, a URI describes:

- The process used to access the content
- The specific computer that stores the content
- The specific name of the content (i.e. the file name) on the computer

URL

Uniform Resource Locator. An HTTP URL can be for any Web page or individual file.

WADL

Web Application Description Language. An XML-based file format that provides a machine-readable description of HTTP-based web applications. These applications are typically REST web services. The purpose of WADL is to allow services on the internet (or any other IP

network) to be described in a machine-processable way, making it easier to create Web 2.0 style applications and create a dynamic way of creating and configuring services. WADL can be thought of as the REST equivalent of Web Services Description Language version 1.1. However, version 2.0 of WSDL can be used to describe REST Web services, thus competing with WADL.

Webroot

In a computer file system organized in a hierarchical or tree structure, the root directory is the directory that includes all other directories (i.e. C:\). For Web sites, the webroot is the uppermost level of the tree hierarchy of the site.

Web form authentication

Many Web applications contain HTML forms that a user must complete successfully before being allowed to access the remainder of the application. Typically, the user types a “user name” in a single-line text input control and “password” in a password control, and then submits the form to a server-based agent for processing.

Web service

Web services are programs that communicate with other applications (rather than with users) and answer requests for information. Most Web services use SOAP (Simple Object Access Protocol) to send XML data between the Web service and the client Web application making 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 WSDL (Web Services Definition Language) document so that it understands how to communicate with the service. The WSDL document describes what programmed procedures the Web service includes, what parameters those procedures expect, and the type of return information the client Web application will receive.

WSDL

Web Service Definition Language. An XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services).

Index

Numerics

15-day trial, 49

A

Abnormal Input, 230

ActiveX, 327

Adaptive Agents, 456

Add Allowed Hosts, 144

Advanced HTTP Parsing, 183

AJAX, 80, 327

Allowed hosts, 181

Allowed Hosts Criteria, 61

AMP. See Assessment Management Platform.

Application Mapping, 439

Application settings

Database, 211

Directories, 212

General, 209

HP Quality Center, 219

IBM rational ClearQuest, 220

License, 212

Logging, 215

Proxy, 215

Reports, 216

Run as a Service, 218

Server Profiler, 213

Smart Update, 219

Step Mode, 215

Support Channel, 219

Assessment Agents, 21

Assessment Management Platform, 47, 95, 96, 218

Attachments, 66, 77, 84

Attack agents, 118, 224, 225, 226, 234

Attack expressions, 204

Attack Info, 78

Audit, 21, 465

Audit Engines, 21, 224, 455

Audit Inputs Editor, 204, 236

Authentication, 155, 255, 284, 465

Authentication methods

Automatic, 133, 191, 259, 297, 311

Digest, 133, 191

HTTP Basic, 191, 259, 297, 311, 450, 467

Integrated Windows, 192

Kerberos, 133, 191, 467

Negotiate, 192

NTLM, 133, 155, 191, 192, 259, 297, 311, 450, 468

Authorization, 465

Auto Fill Web Forms, 144

Automatic authentication, 133, 191, 259, 297, 311

auto upload scans, 96

B

Banner, 465

Base64, 264, 265

Best Practices tab, 87

Blowfish, 265, 266

Broken Links, 81

Brute force, 450

Brute force attack, 439, 450

Button bar, 58

C

Canonicalization, 465

CAPTCHA, 321

Certificates, 80

character frequency, 294

Check Inputs, 237

Client Certificate, 193, 224, 449

Command execution check, 229

Command-line interface, 131, 135

Comments, 23, 77, 80, 99, 450

Compliance Manager, 313

Compress response data, 168

Content analyzers, 172
Content Investigation, 439
Cookie, 23, 291, 440, 442, 443, 445, 449, 451, 466
Cookie Cruncher, 291
Cookie Cruncher settings, 296
Cookies, 80, 327
Copy URL, 83
Crawl, 21, 466
Cross-Site Scripting, 229
Custom Checks, 92, 195, 233, 466
 creating, 227
 definition, 459
Custom Cookies, 190
Customer support, 4
Custom File Not Found, 178
Custom Headers, 189
Custom Parameters, 183
Custom policy, 226

D

Debug information, 167
Default settings
 Audit settings
 Attack Exclusions, 202
 Attack Expressions, 204
 Session Exclusions, 199
 Smart Scan, 205
 Vulnerability Filtering, 204
 Crawl settings
 Link Parsing, 196
 Session Exclusions, 196
 Scan settings
 Allowed Hosts, 181
 Authentication, 192
 Content Analyzers, 172
 Cookies/Headers, 189
 Custom Parameters, 183
 File Not Found, 194
 Filters, 188
 General, 167
 HTTP Parsing, 182
 Method, 165
 Policy, 194
 Proxy, 190
 Recommendations, 173
 Requestor, 174
 Session Exclusions, 178
 Session Storage, 176
DES, 265

Details, 77
Digest authentication, 133, 191, 466
Directory Enumeration, 227
Directory Traversal, 229
Download scan, 95

E

EBCDIC, 265
Edit vulnerability, 66, 84
E-mail, 77, 79, 80, 450
E-mails, 80
Enable Path Truncation, 167
Encoder/Decoder, 264
Engine Inputs, 236
Enterprise Assessment, 151
Evasions, 286
 Case Sensitivity, 288
 DOS/Win Directory Syntax, 288
 Double Slashes, 287
 HTTP Misformatting, 288
 Long URLs, 288
 Method Matching, 286
 NULL Method Processing, 288
 Parameter Hiding, 287
 Reverse Traversal, 287
 Self-Reference Directories, 287
 URL Encoding, 286
Excluded Cookies, 202
Excluded file extension, 177
Excluded Headers, 203
Excluded Hosts, 60
Excluded MIME Types, 178
Excluded URL, 177
Export
 Hosts To Scan list, 154
 log files, 104
 logs, 104
 Macro, 139
 scan, 23, 94, 103, 133
 scan details, 104, 133
 scan settings, 112
 Scan to Software Security Center, 103
 Server Analyzer results, 362
 settings, 113, 206
 site tree, 65
 Web Brute list, 257
Export wizard, 23

F

False positive, 66, 74, 99, 194
File extension addition, 227
File extension replacement, 228
FilesToURLs Utility, 162
Filters, 188, 286
Flag session for follow-up, 66, 84
Flash, 327
Flash files, 172, 175, 283
Forms, 77, 80
Fortify. See HP Fortify.
Fuzzer filters, 300
Fuzzer generators, 299

G

generate report, 118
Generate Session Report, 65
generator, 299
Generators, Web Fuzzer, 299
global form entry, 246
GZIP, 275

H

hexadecimal, 265
Hiddens, 77, 80
Host Info panel, 79
HP Fortify, 77
HP Quality Center, 66, 84, 219
HP SecurityScope, 31, 68, 77, 82, 89, 119, 210
HP Support Tool, 416
HTTP, 182, 467
HTTP Basic authentication, 191, 259, 297, 311, 450, 467
HTTP Editor, 22, 257, 265, 268, 271, 282, 325
HTTP Editor settings, 275
HTTP parsing, 182
HTTP Request, 76
HTTP Response, 76
hyperlinks, 440, 469

I

IBM Rational ClearQuest, 66, 84, 220, 221

IBM WebSphere Portal, 133, 141

Icons, 63, 64, 83, 224, 235, 383
Ignore vulnerability, 86
IIS, 191, 205, 226, 237, 297, 311
import
 Audit Inputs, 204, 236
 check input modifications, 236
 list of proxy servers, 284
 policies, 195
 proxy server information, 190, 252, 259, 278, 297, 306, 311, 327, 361
 scans, 94, 109
 settings, 113, 206
 Web Brute list, 257
 Web form file, 250
include parameters in hit count, 171

Information pane, 67, 97, 99
Information tab, 86
Installation, 45
Integrated Windows authentication, 192
Intelligent engines, 21
Interactive mode, 277, 283, 289

J

Japanese, 307
Java, 266
JavaScript, 23, 172, 173, 209, 229, 250, 275, 442
JavaScript “include” files, 175
JRun, 456

K

Kerberos authentication, 133, 191, 467
Keyword search, 170, 228, 233, 457
Knowledgebase, 290, 439
Known Attacks, 439, 451
Known Vulnerabilities, 457

L

Launch Interactive, 321
License agreement, 47
License Wizard, 48
Links, 65, 77
Listener Configuration, 283
Locations, manually adding, 65
Login macro, 193, 279, 468

Log Viewer, 318

M

Macro

- Login, 468
- Startup, 471
- Web, 281

Manage Scans, 58

Manage Schedule, 58

Manually adding locations, 65

Maximum crawl count, 171

Maximum crawl folder depth, 171

Maximum link traversal sequence, 171

Maximum single URL hits, 170

Maximum URL Hits, 177

MD5, 264, 265

Memo header, 167

Menu bar, 93

Microsoft Excel, 83

MIME type, 178, 197, 468

N

Navigation pane, 59, 97, 99

Negotiate authentication, 192

NTLM authentication, 133, 155, 191, 192, 259, 297, 311, 450, 468

O

Offsite Links, 81

Oracle, 307, 366, 367, 368, 369

Oracle ADF Faces, 40, 133, 141

Outside Root URL, 177

P

P3P Info, 79

Parameter injection, 228

Parameter Manipulation, 439

Parameters, 81

passwords, 99, 255, 465, 467

Path Manipulation, 439

Policies, 453

policy

- editing, 226

Policy Manager, 224

Postdata, 304, 440

Postdata injection, 457

Proxy, 284

Proxy server, 140, 190, 192, 215, 216, 218, 241, 246, 251, 252, 259, 260, 276, 278, 279, 283, 284, 297, 298, 305, 306, 307, 311, 312, 327, 328, 361, 418, 468

Proxy Settings, 190, 192, 272, 276, 278, 279, 281, 298, 306, 310, 312, 319, 361

Q

Quality Center, 66, 84, 219

Query string, 81, 182, 238, 239, 302, 303, 323, 440

R

randomness, 294

RC2, 265

RC4, 265

Recommendations, 32, 36, 67, 71, 173

Redundant page detection, 170

Regular Expression Editor, 267

Regular Expressions, 268

Rejected Response, 178

Remove Server, 66

Remove session, 84

Report Designer, 374

Reports

 generating, 118

Requestor Settings, 175

Request retry count, 175

Request timeout, 175

Rescan, 32, 37, 72, 91, 95, 119

Retest, 33, 34, 66, 69, 83, 84, 85, 98

Retesting vulnerabilities, 85

Retry failures, 176

Review vulnerability, 84, 85

ROT13, 265

S

Scan details, 167

Scan Info panel, 67

Scan Log tab, 87

Scanning policies, 22, 453

Scan toolbar, 90

Schedule, 94
Scheduled scans, 114
Screen capture, 66
Scripts, 77, 81
Scrub Data, 103, 105
Search view, 22, 62
Secure Hash Algorithm, 266
SecurityScope. See HP SecurityScope
Send To, 84, 86, 98, 99
Send to, 66
separate requestors, 175
Sequence view, 22, 76
Server Analyzer, 360
Server Analyzer settings, 360
Server Information tab, 87
Server Profiler, 363
Session Editor, 302
Session exclusions, 178
Session hijacking, 469
session ID, 291, 469
Session Info panel, 76
Session storage, 176
settings
 WebInspect application, 209
 WebInspect default, 165
SHA, 266
SHA-256, 266
SHA-384, 266
SHA-512, 266
shared requestor, 175
Simple attack, 230
Site search, 230
Site Searching, 439
Site view, 22, 59, 76
Smart Assessment, 205
Smart Credentials, 140, 141, 250, 325, 340
Smart Scan, 205
Smart Update, 57, 219, 290, 451, 453
 on startup, 219
Snapshot, 66
SOAP, 23, 470
software installation, 45
Software Security Center, 82, 94, 103, 104
Solicited File Not Found, 178
SQL injection, 229, 307, 470
SQL Injector, 307
SQL Injector settings, 309
Stack trace, 31, 77, 82, 89, 118, 161, 376
Standard toolbar, 92
Start Page, 57
Startup macro, 194, 279, 310, 471
Step Mode, 193, 471
Step Mode (in navigation pane), 63
Steps, 77
Subcookies, 292
Summary pane, 81
Support, 4
SWFScan, 365

T

Text, 77
ToLower, 266
Toolbars, WebInspect, 90
Tools
 Audit Inputs Editor, 236
 Compliance Manager, 313
 Cookie Cruncher, 291
 Encoder/Decoder, 264
 HP Support Tool, 416
 HTTP Editor, 271
 Log Viewer, 318
 Policy Manager, 224
 Regular Expression Editor, 267
 Report Designer, 374
 Server Analyzer, 360
 Server Profiler, 363
 Smart Update, 290
 SQL Injector, 307
 SWFScan, 365
 Web Application Firewall Integration Tool, 436
 Web Brute, 255
 Web Discovery, 261
 Web Form Editor, 246
 Web Fuzzer, 299
 Web Macro Recorder (Event-Based), 330
 Web Macro Recorder (Session-Based), 319
 Web Macro Recorder (TruClient), 342
 Web Proxy, 85, 279
 Web Service Test Designer, 420

Tool settings

- Cookie Cruncher, 296
 HTTP Editor, 275
 Server Analyzer, 360
 SQL Injector, 309
 SWFScan, 372
 Web Brute, 258
 Web Discovery, 262
 Web Form Editor, 251
 Web Fuzzer, 305
 Web Macro Recorder (Session-Based), 326
 Web Proxy, 283
ToUpper, 266
 Traffic Analysis, 144
 Traffic Monitor, 67, 70, 144, 150, 169
 Transfer settings to/from AMP, 96
 TwoFish, 266
- U**
- Unicode, 264, 266
 Upload scan, 96
 URL encoding, 266
- V**
- Variation, 65
 VBScript, 172, 196
 Verbose, 134
 View in browser, 83
 Vulnerabilities tab, 82
 Vulnerability Detail section, 101
 Vulnerability filtering, 204
 Vulnerability snapshot, 66
- W**
- WADL, 183, 184, 185, 471
 Web Application Firewall Integration Tool, 436
 Web Browser, 76
 Web Brute, 255
 Web Brute settings, 258
 Web Discovery, 261
 Web Discovery settings, 262
 Web Form Editor, 246
 Web Form Editor settings, 251
 Web Form list
 - creating manually, 246
 - recording, 248
 Web form submissions, 171
 Web Fuzzer, 299
 Web Fuzzer settings, 305
 WebInspect settings
 - Application settings, 209
 - Default Settings, 165
 Web macro, 281
 Web Macro Recorder (Event-Based), 330
 Web Macro Recorder (Session-Based), 319
 Web Macro Recorder (TruClient), 342
 Web Proxy, 22, 85, 279
 - interactive mode, 289
 Web Proxy settings, 283
 Web Server Assessment, 439
 Web Service Assessment, 148
 Web Service operations, 431
 Web Service Test Designer, 420
 Web Site Assessment, 137
 windows
 - Add Check By ID, 315
 - Add Existing Vulnerability, 100
 - Add Profile, 220
 - Add Request/Response Data Filter Criteria, 188
 - Add Scrub Entry, 103, 105
 - Add User-Defined Input, 247
 - Add Variation, 65
 - Advanced Settings, 142
 - AMP Configuration, 290
 - Application Settings, 50
 - Application settings, 87, 94
 - Browse For Folder, 212
 - Bypass Proxy, 284
 - Certificates, 193
 - Client Certificates, 192
 - Compliance Manager, 313
 - Configure Report Settings, 116
 - Convert Web Form Values, 250
 - Create New Settings, 112, 206
 - Create Report Definition, 375
 - Create Web Macro, 281
 - Current Settings, 61
 - Default Settings, 94, 113, 250
 - Destination Folder, 47
 - Edit Vulnerabilities, 65, 98, 100
 - Edit Web Site Scan, 154
 - Enterprise Scan Wizard, 152
 - Exclusion Extension, 178, 197, 199
 - Export A Scan, 103
 - Export Dictionary, 258
 - Export File, 362

Export Scan Details, 104, 436
Export Scan to Software Security Center, 103,
 104
Filters, 301
Find in Request, 275
Find in Response, 274, 275
Form Values Detection, 73
Generate a Report, 118, 120, 316
HTTP Parameter, 183
Import/Export Dictionary, 257
Import Custom Policy, 195
Import Dictionary, 257
Internet Options, 193
LAN Settings, 279, 283
Logout Condition Editor, 321
Manage Scan Scheduling, 94
Manage Settings, 112, 206
Modify Input, 247
Open a Report, 375
Open Policy, 234
Open Scan, 318
Open Scan Settings File, 206
Properties, 397
Provide a Mime-type to Exclude, 179, 197, 200
Ready to Install, 48
Recommendations, 71
Recover Deleted Items, 69, 84, 85
Regular Expression Editor, 267
Reject or Exclude a Host or URL, 179, 197, 200
Review Vulnerability, 85
Run Scan in AMP, 110
Save As, 258
Save Scan Settings, 206
Scan Comparison, 158
Scan Wizard, 137
Search for Web Servers, 153
Select a Report, 116
Select Data Dictionary, 256
Sensor Configuration, 47
Server/Application Type Entry, 205
Session Properties, 154
Settings Properties, 262
Smart Update, 290
Specialized Link Entry, 196
Specify Allowed Host, 144, 181
Specify Custom Cookie, 190
Specify Custom Header, 189
Specify HTTP Exclusions, 202
Test Login Macro, 320, 329
Vulnerability Compare, 86
Vulnerability Review, 86, 156, 162
WebForm Editor, 247
Web Fuzzer, 302
Web Fuzzer Request, 300
WebInspect, 59, 90

WebInspect Product Registration Wizard, 49
Web Proxy, 279
Web Proxy Settings, 289
Web Service Scan Wizard, 148

WSDL, 472

X

XML, 23, 99, 206, 458, 470, 472
XML Reques, 78
XML Response, 78
XOR, 266

Z

zero.webappsecurity.com, 246
zlib, 275

