opentext™

OpenText™ Dynamic Application Security Testing (Fortify WebInspect)

Software Version: 26.4.0
Windows® operating systems

User Guide

Document Release Date: October 2026
Software Release Date: October 2026

Legal notices

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Copyright notice

Copyright 2004-2026 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Trademark notices

“OpenText” and other Open Text trademarks and service marks are the property of Open Text or its affiliates. All other trademarks or service marks are the property of their respective owners.

Documentation updates

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

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

This document was produced for OpenText™ Dynamic Application Security Testing on October 05, 2026.

About this PDF Version of Online Help

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

Contents

Preface ... 24
Contacting Customer Support ... 24
For more information ... 24
Product feature videos ... 24

Change log ... 25

Chapter 1: Introduction ... 27
Understanding the findings ... 27
OpenText DAST overview ... 27
Crawling and auditing ... 27
Reporting ... 28
Manual hacking control ... 28
Summary and fixes ... 28
Scanning policies ... 28
Sortable and customizable views ... 29
Enterprise-wide usage capabilities ... 29
Web services scan capabilities ... 29
Export wizard ... 29
API scans ... 29
API discovery ... 30
Integration capabilities ... 30
Enhanced third-party commercial application threat agents ... 30
Hacker-level insights ... 30
About FIPS compliance in OpenText DAST products ... 31
Product name changes ... 31

Chapter 2: Getting started ... 32
Preparing your system for audit ... 32
Sensitive data ... 32
Firewalls, anti-virus software, and intrusion detection systems ... 32
Effects to consider ... 33
Helpful hints ... 33
Quick start ... 34

Update SecureBase 35
Prepare your system for audit 35
Start a scan 35

Chapter 3: OpenText DAST user interface 36
The Activity Panel 36
Closing the Activity Panel 37
The Button Bar 37
Panes Associated with a Scan 37
Start Page 38
Home 38
Manage Scans 38
Manage Schedule 39
About the menu bar 39
File menu 39
Edit menu 40
View menu 41
Tools menu 41
Scan menu 41
Reports menu 42
Help menu 43
DAST Help 43
Support 43
Tutorials 43
DAST Community 43
About DAST 43
Toolbars 43
Buttons available on the Scan toolbar 43
Buttons available on the Standard toolbar 44
Buttons available on the Manage Scans toolbar 45
Navigation pane 46
Site view 48
Excluded hosts 48
Allowed hosts criteria 49
Sequence View 50
SPA coverage 51

Search view ... 52
Step Mode view ... 53
Navigation pane icons ... 53
Navigation pane shortcut menu ... 55
Information pane ... 57
Scan Info panel ... 58
Dashboard ... 58
Traffic Monitor ... 59
Attachments ... 59
Suppressed Findings ... 60
Dashboard ... 60
Progress bars ... 61
Progress bar descriptions ... 61
Progress bar colors ... 62
Activity meters ... 62
Activity meter descriptions ... 63
Findings graphics ... 63
Statistics panel - Scan ... 63
Statistics panel - Crawl ... 65
Statistics panel - Audit ... 65
Statistics panel - Network ... 65
Attachments - Scan Info ... 66
Suppressed findings ... 67
Understanding suppressed findings ... 67
Importing suppressed findings ... 67
Inactive / Active Suppressed Findings lists ... 68
Loading suppressed findings during scan configuration ... 68
Working with suppressed findings ... 68
Session Info panel ... 69
Options available ... 70
Vulnerability ... 73
Web browser ... 73
HTTP Request ... 73
Highlighted text in the request ... 73
HTTP Response ... 74
Highlighted text in the response ... 74
Stack traces ... 74
Details ... 74
Steps ... 74
Links ... 75

Comments - session info ... 75
Text ... 75
Hiddens: Session Info ... 75
Forms: Session Info ... 76
E-mail ... 76
Scripts - Session Info ... 76
Attachments - Session Info ... 76
Viewing an attachment ... 77
Adding a session attachment ... 77
Editing an attachment ... 77
Attack Info ... 78
Web service request ... 78
Web service response ... 78
XML request ... 78
XML response ... 78
Host Info panel ... 79
Options available ... 79
P3P info ... 80
P3P user agents ... 81
AJAX ... 81
How AJAX works ... 82
Certificates ... 83
Comments - host info ... 83
Cookies ... 83
E-mails - Host Info ... 84
Forms - Host Info ... 84
Hiddens - Host Info ... 85
Scripts - Host Info ... 85
Broken links ... 86
Offsite links ... 86
Parameters ... 86
Summary pane ... 87
Findings tab ... 88
Available columns ... 89
Vulnerability severity ... 90
Working with findings ... 90
Scan Log tab ... 92
Server Information tab ... 93
OpenText DAST Monitor ... 93

Chapter 4: Working with scans 95
Guided Scan overview 95
Predefined templates 95
Mobile templates 95
Running a Guided Scan 96
Predefined template (Standard, Quick, or Thorough) 96
Mobile scan template 96
Native scan template 97
Using the Predefined template 97
Recommendation 97
Launching a Guided Scan 97
Understanding the Rendering Engine 98
Verifying your website 98
Choosing a scan type 100
Configuring network authentication 101
Using a client certificate 102
Configuring application authentication 102
Masked values supported 103
Using a login macro without privilege escalation 103
Using login macros for privilege escalation 104
Automatically creating a login macro 105
Configuring workflows 105
Adding Burp Proxy results 106
Using the Profiler 106
Configuring additional options 109
Validating settings and starting the scan 109
Using the Mobile Scan template 110
Recommendation 110
Launching a Mobile Scan 110
Creating a custom user agent header 111
Verifying your website 111
Choosing a scan type 113
Configuring network authentication 114
Using client certificates 115
Configuring application authentication 116
Masked values supported 116
Using a login macro without privilege escalation 116
Using login macros for privilege escalation 117
Automatically creating a login macro 118
About the Workflows stage 118

Adding Burp Proxy results 119
Using the Profiler 120
Configuring additional options 122
Validating settings and starting the scan 122
Using the Native Scan template 123
Recommendation 123
Setting up your mobile device 123
Understanding Guided Scan stages 123
Supported devices 124
Supported development emulators 124
Launching a Native Scan 124
Choosing device/emulator type 125
Selecting a profile 125
Setting the mobile device proxy address 125
Adding a trusted certificate 126
Choosing the scan type 127
Configuring network authentication 128
Using a client certificate 128
Configuring application authentication 129
Masked values supported 130
Using a login macro without privilege escalation 130
Using login macros for privilege escalation 130
Testing the macro 132
Running the application 132
Finalizing allowed hosts and RESTful endpoints 132
Reviewing settings 133
Validating settings and starting the scan 134
Post scan steps 134
Importing Functional Testing files in a Guided Scan 135
Running an API or web service scan 136
Important information about SOAP web service scans 136
Important information about gRPC proto files 137
Known limitations of gRPC scans 137
Options for conducting scans 137
Using the API Scan Wizard 137
API scans 137
Web service scans 138
Recommendation 138
Getting started with the API Scan Wizard 138
What's next? 138

Configuring an API scan ... 139
Configuring authentication and connectivity for API scans ... 142
Configuring proxy settings for API and web service scans ... 142
Configuring network authentication for API and web service scans ... 143
Fetching a token value ... 144
Using a client certificate ... 145
Updating certificates in composite scan settings ... 145
Using custom headers ... 146
Configuring SOAP authentication ... 147
What's next? ... 148
Configuring API content and filters ... 148
Viewing and adjusting Postman configuration settings ... 148
Specifying the preferred content type ... 149
Defining specific operations to include ... 150
Defining specific operations to exclude ... 150
Editing specific operations ... 150
Removing specific operations ... 151
Defining parameter rules ... 151
Editing a parameter rule ... 153
Removing a parameter rule ... 153
What's next? ... 153
Understanding parameter type matches ... 153
Configuring API audit coverage and thoroughness ... 154
Selecting one or more policies for API scans ... 154
What's next? ... 155
Saving settings or starting the API scan ... 155
Saving settings ... 155
Starting a scan ... 155
Scanning an API with wi.exe ... 155
Process overview ... 156
Important considerations about definition files ... 157
Recommendations ... 157
Understanding the API scan configuration file ... 157
Understanding parameter rule objects ... 160
Understanding API AuthProviders configuration ... 163
Client certificate ... 163
Message based ... 164
Transport ... 166
Transport custom ... 168
Fetching a token value ... 169
API scan configuration file samples ... 170

Sample GraphQL configuration file ... 170
Sample gRPC configuration file ... 170
Sample SOAP configuration file ... 171
Running a Basic Scan (website scan) ... 171
Recommendation ... 172
Configuring Basic Scan options ... 172
Configuring network authentication and connectivity ... 175
Configuring proxy settings ... 175
Configuring network authentication ... 176
Using client certificates ... 176
Updating certificates in composite scan settings ... 177
Configuring site authentication ... 178
Configuring crawl coverage and thoroughness ... 179
Configuring audit coverage and thoroughness ... 181
Using the Profiler ... 182
Choosing Profiler suggested settings ... 182
Auto fill web forms ... 183
Add allowed hosts ... 183
Reuse identified Suppressed Findings ... 183
Sample macro ... 184
Traffic analysis ... 184
Congratulations ... 185
Saving settings ... 185
Generating reports ... 185
Using the Site List Editor ... 186
Configuring the proxy profile ... 186
Configure proxy using a PAC file ... 187
Explicitly configure proxy ... 187
Specifying allowed hosts ... 187
Specifying allowed hosts ... 188
Editing allowed hosts ... 188
Multi-user login scans ... 189
Before you begin ... 189
Known limitations ... 189
Process overview ... 190
Configuring a multi-user login scan ... 190
Adding credentials ... 191
Editing credentials ... 192
Deleting credentials ... 192
Using two-factor authentication ... 192

How scanning with two-factor authentication works 192
Recommendation 193
Known limitations 193
Facts about Gmail accounts 193
Understanding the process 193
Interactive scans 194
Configuring an interactive scan 195
Restrict to folder limitations 196
JavaScript include files 197
Login macros 197
Workflow macros 197
Running a manual scan 197
About privilege escalation scans 198
Two modes of privilege escalation scans 199
What to expect during the scan 199
Regex patterns used to identify restricted pages 199
Modifying regex for privilege restriction patterns 200
Effect of crawler limiting settings on privilege escalation scans 200
Effect of parameters with random numbers on privilege escalation scans 201
About single-page application scans 201
The challenge of single-page applications 202
Enabling SPA support 202
Scan status 202
Updates to information in the scan manager 203
Opening a saved scan 204
Comparing scans 204
Selecting scans to compare 205
Reviewing the Scan Dashboard 206
Compare modes 207
Session filtering 207
Using the Session Info panel 208
Using the Summary Pane to review vulnerability details 208
Manage Scans 210
Schedule a scan 212
Configuring time interval for scheduled scan 213
Managing scheduled scans 213
Accessing scheduled scans 214

Deleting a scan ... 214
Editing scan settings ... 214
Running a scan immediately ... 214
Stopping a scheduled scan ... 214
Scheduling a scan ... 215
Selecting a report ... 215
Configuring report settings ... 216
Stopping and restarting a scheduled scan ... 216
Scheduled scan status ... 217
Exporting a scan ... 217
Exporting scan details ... 219
Export scan to Application Security Center ... 221
Known issue with uploading FPRs from paused scans ... 222
Exporting protection rules to Web Application Firewall (WAF) ... 223
Importing a scan ... 223
Selecting a scan to import suppressed findings ... 224
Importing legacy web service scans ... 224
Importing and exporting scan settings ... 225

Chapter 5: Using OpenText DAST features ... 226
Retesting and rescanning ... 226
Retesting vulnerabilities ... 226
Understanding the retest status ... 227
Recommendation for failed and not supported vulnerabilities ... 228
Retesting all vulnerabilities ... 228
Retesting all vulnerabilities with a specific severity ... 228
Retesting selected vulnerabilities ... 229
Retesting grouped categories ... 229
Retesting a retest scan ... 230
Retest scan log ... 230
Comparison views ... 230
Keeping or deleting a retest scan ... 230
Rescanning a site ... 231
Reusing scans ... 232
Reuse options ... 232
Difference between remediation scans and retest vulnerability ... 232
Guidelines for reusing scans ... 232

Reusing a scan ... 232
Incremental scanning ... 233
Merging baseline and incremental scans ... 233
Incremental with continuous audit ... 234
Incremental with deferred audit ... 234
Using macros ... 235
Selecting a workflow macro ... 236
Using a Web Macro Recorder ... 236
Event-based Web Macro Recorder ... 236
Session-based Web Macro Recorder ... 237
Traffic Monitor (Traffic Viewer) ... 237
Traffic session data in Traffic Viewer ... 237
Viewing traffic in the Traffic Viewer ... 237
Server Profiler ... 238
Launching Server Profiler as a tool ... 238
Invoke Server Profiler when starting a scan ... 239
Inspecting the results ... 239
Working with one or more vulnerabilities ... 240
Working with a Group ... 241
Understanding the Severity ... 241
Working in the Navigation pane ... 242
Client-side library analysis ... 242
NVD information ... 242
Debricked health metrics ... 243
Debricked content contingent upon access ... 243
Search view ... 244
Using filters and groups in the Summary pane ... 244
Using filters ... 245
No filters ... 245
Filtered by Method:Get ... 245
Specifying multiple filters ... 246
Filter criteria ... 246
Using groups ... 247
Auditing web services ... 247
Options Available from the Session Info Panel ... 248
Adding/viewing vulnerability screenshot ... 250
Viewing screenshots for a selected session ... 250
Viewing screenshots for all sessions ... 250

Editing vulnerabilities 251
Editing a vulnerable session 252
Vulnerability rollup 253
What happens to rolled up vulnerabilities 253
Rollup guidelines 253
Rolling up vulnerabilities 254
Undoing rollup 254
Mark as false positive 255
Mark as vulnerability 255
Flag session for follow-up 255
Viewing flags for a selected session 256
Viewing flags for all sessions 256
Using scan notes 256
Working with session notes 257
Viewing notes for a selected session 257
Viewing notes for all sessions 257
Vulnerability note 257
Viewing notes for a selected session 258
Viewing notes for all sessions 258
Recovering deleted sessions 258
Sending vulnerabilities to OpenText ALM 259
Additional information sent 259
Disabling Data Execution Prevention 260
Generating a report 260
Saving a report 261
Advanced report options 261
Report viewer 262
Adding a note 263
Standard reports 263
Manage reports 265
Compliance templates 265
Managing settings 277
Accessing the Manage Settings window 278
Creating a Settings File 278
Editing a Settings File 278
Deleting a Settings File 278

Importing a Settings File 278
Exporting a Settings File 279
Scanning with a Saved Settings File 279
SmartUpdate 279
Performing a SmartUpdate (Internet connected) 280
Downloading checks without updating OpenText DAST 280
Performing an offline SmartUpdate 281
WebSphere Portal FAQ 281
Command-line execution 283
Launching the CLI 283
CLI limitations in OpenText DAST on Docker 284
Using wi.exe 284
Options 284
Examples 301
Selenium login macro example 301
Response state rule example 301
Merging scans 302
Hyphens in command line arguments 302
Exit codes 302
Using WIScanStopper.exe 303
Using MacroGenServer.exe 303
Options 304
Using the WISwag.exe tool 305
Supported API definitions and protocols 305
Locating the WISwag.exe tool 305
Process overview 305
WISwag.exe parameters 306
Converting the API definition to a macro 308
Converting the API definition to a settings file 308
Using a configuration file 308
Configuration file format 309
Regular expressions 310
Regex extensions 311
Regular expression tags 311
Regular expression operators 312
Examples 312
OpenText DAST REST API 313
What is the OpenText DAST REST API? 313
Recommendation 313

Configuring the OpenText DAST REST API ... 314
Accessing the OpenText DAST API Swagger UI ... 316
Switching between API versions ... 317
Using the Swagger UI ... 317
Getting field-level details ... 318
Automating OpenText DAST ... 319
OpenText DAST updates and the API ... 319
Scanning with a Postman collection ... 319
What is Postman? ... 319
Benefits of a Postman collection ... 320
Known limitations with Postman variables ... 320
Options for Postman scans ... 320
Postman prerequisites ... 320
Using client certificates with Postman ... 320
Tips for preparing a Postman collection ... 321
Ensure valid responses ... 321
Order of requests ... 321
Handling authentication ... 321
Using static authentication ... 322
Using dynamic authentication ... 322
Using a Postman login macro ... 322
Postman auto-configuration ... 322
Sample Postman scripts ... 323
Manually configuring Postman login for dynamic tokens ... 323
What are dynamic tokens? ... 323
Before you begin ... 323
Process overview ... 323
Identifying and isolating the login request ... 324
Creating a logout condition with regular expressions ... 324
Creating a response state rule for a bearer token ... 325
Creating a response state rule for an API key ... 325
Postman API scan using WI.exe or OpenText DAST REST API ... 326
Recommendation ... 326
Process ... 327
Troubleshooting a Postman scan ... 328
Integrating with Selenium WebDriver ... 328
Known limitations ... 328
Process overview ... 329
Adding the proxy to Selenium scripts ... 330
Advantages ... 330

Disadvantages ... 330
Sample code ... 330
Using the CLI ... 335
Using the OpenText DAST geckodriver.exe ... 335
Advantages ... 335
Disadvantages ... 335
Installing the Selenium WebDriver environment ... 335
Testing from the command line ... 336
Creating a Selenium command ... 336
Uploading files to OpenText DAST ... 339
Using the CLI ... 339
Using the API ... 339
Using the Selenium command ... 340
Running a scan using WI.exe ... 340
Creating a macro using the API ... 341
About the Burp API extension ... 342
Benefits of using the Burp API extension ... 342
Supported versions ... 342
Using the Burp API extension ... 343
Loading the Burp extension ... 343
Connecting to OpenText DAST ... 344
Refreshing the list of scans ... 346
Working with a scan in Burp ... 346
Sending items from Burp to OpenText DAST ... 349
About the WebInspect SDK ... 350
Audit extensions / custom agents ... 351
SDK functionality ... 351
Installation recommendation ... 351
Installing the WebInspect SDK ... 352
Verifying the installation ... 352
After installation ... 353
Add page or directory ... 353
Add variation ... 354
Creating an exclusion ... 354
Example 1 ... 355
Example 2 ... 355
Example 3 ... 356
Example 4 ... 356
Internet Protocol version 6 ... 356

Chapter 6: Default scan settings 358
Scan settings: Method 358
Scan mode 358
Crawl and audit mode 359
Crawl and audit details 359
Navigation 360
SSL/TLS protocols 361
Scan settings: General 361
Scan details 362
Crawl details 363
Scan settings: JavaScript 367
JavaScript settings 367
Scan settings: Requestor 369
Requestor performance 369
Requestor settings 370
Stop scan if loss of connectivity detected 371
Scan settings: Session Exclusions 372
Excluded or rejected file extensions 372
Excluded MIME types 373
Other exclusion/rejection criteria 373
Editing criteria 373
Adding criteria 374
Scan settings: Allowed Hosts 375
Using the Allowed Host setting 375
Adding allowed domains 376
Editing or removing domains 376
Scan settings: HTTP Parsing 376
Options 377
CSRF 381
About CSRF 381
Using CRSF tokens 382
Enabling CSRF awareness in OpenText DAST 382
Scan settings: Custom Parameters 382
URL rewriting 382
RESTful services 383
Working with rules 383
Enable automatic seeding of rules that were not used during scan 384
Double encode URL parameters 384

Path matrix parameters ... 385
Definition of path segment ... 385
Special elements for rules ... 386
Asterisk placeholder ... 387
Benefit of using placeholders ... 388
Multiple rules matching a URL ... 388
Scan settings: Filters ... 388
Options ... 389
Adding rules for finding and replacing keywords ... 389
Scan settings: Cookies/Headers ... 390
Standard header parameters ... 390
Append custom headers ... 390
Adding a custom header ... 391
Append custom cookies ... 391
Adding a custom cookie ... 391
Scan settings: Proxy ... 392
Options ... 392
Scan settings: Authentication ... 393
Scan requires network authentication ... 393
Authentication method ... 394
Authentication credentials ... 394
Client certificates ... 394
Updating certificates in composite scan settings ... 395
Editing the proxy config file for OpenText DAST tools ... 396
Enable macro validation ... 397
Use a login macro for forms authentication ... 397
Login macro parameters ... 398
Use a startup macro ... 398
Multi-user login ... 398
Configuring OAuth 2.0 bearer credentials ... 400
Scan settings: File Not Found ... 401
Options ... 401
Scan settings: Policy ... 403
Selecting one or more policies ... 403
Creating a policy ... 403
Editing a policy ... 403
Importing a policy ... 404
Deleting a policy ... 404

Scan settings: User Agent ... 404
Profile and user-agent string ... 405
Navigator interface settings ... 406

Chapter 7: Crawl settings ... 408
Crawl settings: DOM Coverage ... 408
Setting a DOM Coverage limit ... 408
Setting a DOM Coverage strategy ... 408
Crawl settings: Link Parsing ... 409
Adding a specialized link identifier ... 409
Crawl settings: Link Sources ... 409
What is link parsing? ... 409
Pattern-based parsing ... 410
DOM-based parsing ... 410
Form actions, script includes, and stylesheets ... 415
Miscellaneous options ... 415
Limitations of link source settings ... 416
Crawl settings: Session Exclusions ... 417
Excluded or rejected file extensions ... 417
Adding a file extension to exclude/reject ... 417
Excluded MIME types ... 417
Adding a MIME type to exclude ... 417
Other exclusion/rejection criteria ... 418
Editing the default criteria ... 418
Adding exclusion/rejection criteria ... 418

Chapter 8: Audit settings ... 421
Audit settings: Session Exclusions ... 421
Excluded or rejected file extensions ... 421
Adding a file extension to exclude/reject ... 421
Excluded MIME types ... 421
Adding a MIME type to exclude ... 422
Other exclusion/rejection criteria ... 422
Editing the default criteria ... 422
Adding exclusion/rejection criteria ... 422
Audit settings: Attack Exclusions ... 424
Excluded parameters ... 424
Adding parameters to exclude ... 425
Excluded cookies ... 425

Excluding certain cookies ... 425
Excluded headers ... 426
Excluding certain headers ... 426
Audit Inputs Editor ... 427
Audit settings: Attack Expressions ... 427
Additional regular expression languages ... 427
Audit settings: Vulnerability Filtering ... 427
Adding a vulnerability filter ... 428
Suppressing off-site vulnerabilities ... 428
Audit settings: Smart Audit ... 429
Enable Smart Audit ... 429
Use regular expressions on HTTP responses ... 429
Use server analyzer fingerprinting and request sampling ... 429
Custom server/application type definitions ... 429

Chapter 9: Application settings ... 431
Application settings: General ... 431
General ... 431
OpenText DAST Agent ... 434
Application settings: Database ... 435
Connection settings for scan/report storage ... 435
SQL Server database privileges ... 435
Configuring SQL Server Standard Edition ... 435
Configuring PostgreSQL ... 436
Connection settings for scan viewing ... 436
Application settings: Directories ... 437
Changing where OpenText DAST files are saved ... 437
Application settings: License ... 437
License details ... 437
Direct connection to OpenText ... 438
Connection to LIM ... 438
Application settings: Server Profiler ... 439
Modules ... 439
Application settings: Step Mode ... 441
Application settings: Two-Factor Authentication ... 442
Two-factor authentication control center ... 442
Mobile application ... 442
Installing and configuring the Fortify2FA mobile app ... 443

Application settings: Logging ... 450
Application settings: Proxy ... 450
Not using a proxy server ... 450
Using a proxy server ... 451
Configuring a proxy ... 451
Application settings: Reports ... 452
Options ... 452
Headers and footers ... 453
Application settings: override SQL database settings ... 454
Override database settings ... 454
Configure SQL Database ... 454
Application settings: Smart Update ... 454
Options ... 455
Selecting a different language ... 455
Application settings: Support Channel ... 456
Opening the support channel ... 456
Application settings: OpenText ALM ... 456
ALM License Usage ... 456
Before You Begin ... 456
Creating a Profile ... 457

Chapter 10: Reference lists ... 458
OpenText DAST policies ... 458
About OAST-related checks ... 458
Best Practices ... 458
By Type ... 460
Custom ... 462
Hazardous ... 462
Deprecated Checks and Policies ... 462
Scan log messages ... 464
HTTP status codes ... 487

Chapter 11: Troubleshooting ... 491
Troubleshooting OpenText DAST ... 491
Connectivity issues ... 491
Scan initialization failures ... 492
Scan configuration issues ... 493

Troubleshooting alerts 493
Disabling alerts 493
Alerts troubleshooting table 493
Testing login macros 494
Validation tests performed 495
Troubleshooting tips 495
Uninstalling OpenText DAST 496
Options for removing 497

Send Documentation Feedback 498

Preface

Contacting Customer Support

Visit the Customer Support website to:

- Manage licenses and entitlements
- Create and manage technical assistance requests
- Browse documentation and knowledge articles
- Download software
- Explore the Community

For more information

For more information about OpenText Application Security Testing products, visit OpenText Application Security.

Product feature videos

You can find videos that highlight OpenText Application Security Software products and features on the Fortify Unplugged YouTube™ channel.

Change log

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

Software Release / Document Version	Changes
26.4.0	Updated: • DOM Coverage added to Crawl Settings • Smart Scan renamed to Smart Audit •
26.2.0	Updated: • UI images to show OpenText product name and branding.
25.4.0	Updated: • Default scan settings with new Firefox browser version in user agent list. See "Scan settings: User Agent" on page 404.
25.2.0	Updated: • Fortify WebInspect references to OpenText DAST to reflect product name change. • Scan wizard, wi.exe, and scan settings content for multi-policy selection. See the following topics: • "Dashboard" on page 60 • "Running a Basic Scan (website scan)" on page 171 • "Configuring API audit coverage and thoroughness" on page 154 (New Topic) • "Using wi.exe" on page 284 • "Scan settings: Policy " on page 403 • Default scan settings with new user agents. See "Scan settings: User Agent" on page 404.

Software Release / Document Version	Changes
	• Application settings with new composite settings option. See "Application settings: General" on page 431. • Details about updating certificates in composite scan settings. See the following topics: • "Running a Basic Scan (website scan)" on page 171 • "Configuring authentication and connectivity for API scans" on page 142 • "Scan settings: Authentication" on page 393

Chapter 1: Introduction

OpenText™ Dynamic Application Security Testing (DAST) 26.4.0 is an automated web application, API, and web services vulnerability scanning tool. OpenText DAST delivers the latest evolution in scan technology–a web application security product that adapts to any business environment. As you initiate a scan, OpenText DAST assigns agents that dynamically catalog all areas of a web application. These agents report their findings to a main security engine that analyzes the results. OpenText DAST then launches "threat agents" to evaluate the gathered information and apply attack algorithms to determine the existence and relative severity of potential vulnerabilities. With this smart approach, OpenText DAST continuously applies appropriate scan resources that adapt to your specific application environment.

Understanding the findings

You should consider OpenText DAST findings to be potential vulnerabilities rather than actual vulnerabilities. Every application is unique, and all functionality runs within a particular context that is understood best by the development team. No technology can fully determine whether a suspect behavior can be considered a vulnerability without direct developer confirmation.

See also

"OpenText DAST overview" below

OpenText DAST overview

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

Crawling and auditing

OpenText DAST uses two basic modes to uncover your security weaknesses.

- A crawl is the process by which OpenText DAST identifies the structure of the target website. In essence, a crawl runs until it can access no more links on the URL.
- An audit is the actual vulnerability scan. A crawl and an audit, when combined into one function, is termed a scan.

Reporting

Use OpenText DAST reports to gain valuable, organized application information. You can customize report details, decide what level of information to include in each report, and gear the report for a specific audience. You can also save any customized report as a template, which you can then use to generate a report using the same reporting criteria, but with updated information. You can save reports in either PDF, HTML, Excel, Raw, RTF, or text format, and you can include graphic summaries of vulnerability data.

Manual hacking control

With OpenText DAST, you can see what's really happening on your site, and simulate a true attack environment. OpenText DAST functionality enables you to view the code for any page that contains vulnerabilities, and make changes to server requests and resubmit them instantly.

Summary and fixes

The information pane displays all summary and fix information for the vulnerability selected in either the navigation pane or the summary pane. For more information, see "Navigation pane" on page 46 and "Summary pane" on page 87.

It also cites reference material and provides links to patches, instructions for prevention of future problems, and vulnerability solutions. Because new attacks and exploits are formulated daily, OpenText frequently updates the summary and fix information database. Use Smart Update on the OpenText DAST toolbar to update your database with the latest vulnerability solution information, or check for updates automatically on startup. For more information, see "SmartUpdate" on page 279 and "Application settings: Smart Update" on page 454.

Scanning policies

You can edit and customize scanning policies to suit the needs of your organization, reducing the amount of time it takes for OpenText DAST to complete a scan. For more information on how to configure OpenText DAST policies, see the Policy Manager help or the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Sortable and customizable views

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

- Site view presents the hierarchical file structure of the scanned site, as determined by OpenText DAST. It also displays, for each resource, the HTTP status code returned by the server and the number of vulnerabilities detected.
- Sequence view displays server resources in the order OpenText DAST encountered them during an automated scan or a manual crawl (Step Mode).
- Search view enables you to locate sessions that match the criteria you specify. For more information, see "Search view" on page 244.
- 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. For more information, see "Running a manual scan" on page 197.

Enterprise-wide usage capabilities

Integrated scan provides a comprehensive overview of your web presence from an overall enterprise perspective, enabling you to conduct application scans of all web-enabled applications on the network.

Web services scan capabilities

Provides a comprehensive scan of your web services vulnerabilities. Enables you to assess applications that contain web services/SOAP objects.

Export wizard

OpenText DAST's robust and 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.

API scans

OpenText DAST supports scanning REST API applications as follows:

- Configure an API Scan in the user interface by way of the API Scan Wizard. For more information, see "Using the API Scan Wizard" on page 137.
- Scan a REST API definition using the OpenText DAST REST API. For more information, see "OpenText DAST REST API" on page 313.

- Use a Postman collection of API requests to start a scan. For more information, see "Scanning with a Postman collection" on page 319.

API discovery

With API discovery, any Swagger or OpenAPI schema that is detected during a scan will have its endpoints added to the existing scan and authentication will be applied to the endpoints using automatic state detection. In addition, probes will be sent to default locations of popular API frameworks to discover schemas.

Integration capabilities

You can integrate OpenText DAST with some of the most widely used application security development and testing tools, including the following:

- Burp (For more information, see "About the Burp API extension" on page 342.)
- Postman (For more information, see "Scanning with a Postman collection" on page 319.)
- Selenium WebDriver (For more information, see "Integrating with Selenium WebDriver" on page 328.)
- HTTP Archive (HAR) file workflow macros (For more information, see "Selecting a workflow macro " on page 236.)

Enhanced third-party commercial application threat agents

OpenText DAST enables users to perform security scans for any web application, including the industry-leading application platforms. Some standard commercial application threat agents with OpenText DAST include:

- Adobe ColdFusion
- Adobe JRun
- Apache Tomcat
- IBM Domino
- IBM WebSphere
- Microsoft.NET
- Oracle Application Server
- Oracle WebLogic

Hacker-level insights

OpenText DAST flags libraries that are detected in the application during the scan. This information provides developers and security professionals with context relating to the overall security posture of their application. While these findings do not necessarily

represent a security vulnerability, it is important to note that attackers commonly perform reconnaissance of their target in an attempt to identify known weaknesses or patterns. For more information, see "Client-side library analysis" on page 242.

About FIPS compliance in OpenText DAST products

OpenText DAST programs meet the encryption standards required to be compliant with Federal Information Processing Standard (FIPS). Data is encrypted using the AES algorithm established by the National Institute of Standards and Technology (NIST). This includes the transmission of data to and from OpenText DAST as well as saved scan data.

Product name changes

OpenText is in the process of changing the following product names:

Previous name	New name
Fortify Static Code Analyzer	OpenText™ Static Application Security Testing (OpenText SAST)
Fortify Software Security Center	OpenText™ Application Security
Fortify WebInspect	OpenText™ Dynamic Application Security Testing (OpenText DAST)
Fortify on Demand	OpenText™ Core Application Security
Debricked	OpenText™ Core Software Composition Analysis (OpenText Core SCA)
Fortify Applications and Tools	OpenText™ Application Security Tools

The product names have changed on product splash pages, mastheads, login pages, and other places where the product is identified. The name changes are intended to clarify product functionality and to better align the Fortify Software products with OpenText. In some cases, such as on the documentation title page, the old name might temporarily be included in parenthesis. You can expect to see more changes in future product releases.

Chapter 2: Getting started

This chapter describes how to prepare your system for audit, update SecureBase, and start a scan so that you begin using OpenText DAST right away.

Preparing your system for audit

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

Sensitive data

OpenText DAST captures and displays all application data sent between the application and server. It might even discover sensitive data in your application that you are not aware of. OpenText recommends that you follow one of these best practices regarding sensitive data:

- Do not use potentially sensitive data, such as real user names and passwords, while testing with OpenText DAST.
- Do not allow OpenText DAST scans, related artifacts, and data stores to be accessed by anyone unauthorized to access potentially sensitive data.

Network authentication credentials are not displayed in OpenText DAST and are encrypted when stored in settings.

Firewalls, anti-virus software, and intrusion detection systems

OpenText DAST sends attacks to servers, and then analyzes and stores the results. Web application firewalls (WAF), anti-virus software, firewalls, and intrusion detection/prevention systems (IDS/IPS) are in place to prevent these activities. Therefore, these tools can be problematic when conducting a scan for vulnerabilities.

First, these tools can interfere with OpenText DAST's scanning of a server. An attack that OpenText DAST sends to the server can be intercepted, resulting in a failed request to the server. If the server is vulnerable to that attack, then a false negative is possible.

Second, results or attacks that are in the OpenText DAST product, cached on disk locally, or in the database can be identified and quarantined by these tools. When

working files used by OpenText DAST or data in the database are quarantined, OpenText DAST can produce inconsistent results. Such quarantined files and data can also cause unexpected behavior.

These types of issues are environmentally specific, though McAfee IPS is known to cause both types of problems, and any WAF will cause the first problem. OpenText has seen other issues related to these tools as well.

If such issues arise while conducting a scan, OpenText recommends that you disable WAF, anti-virus software, firewall, and IDS/IPS tools for the duration of the scan. Doing so is the only way to be sure you are getting reliable scan results.

Effects to consider

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

To conduct a thorough scan, OpenText DAST attempts to identify every page, form, file, and folder in your application. If you select the option to submit forms during a crawl of your site, OpenText DAST will complete and submit all forms it encounters. Although this enables OpenText DAST 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), OpenText DAST will also generate these messages as part of its probe.
- If normal form submission causes records to be added to a database, then the forms that OpenText DAST submits will create spurious records.

During the audit phase of a scan, OpenText DAST resubmits forms many times, manipulating every possible parameter to reveal problems in the applications. This greatly increases the number of messages and database records created.

Helpful hints

- For systems that write records to a back-end server (database, LDAP, and so on) based on forms submitted by clients, some OpenText DAST users, before auditing their production system, backup their database, and then restore it after the audit is complete. If this is not feasible, you can query your servers after the audit to search for and delete records that contain one or more of the form values submitted by OpenText DAST. You can determine these values by opening the Web Form Editor tool. For more information, see the Web Form Editor chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
- If your system generates e-mail messages in response to user-submitted forms, 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 OpenText DAST.

- OpenText DAST can be configured to send up to 75 concurrent HTTP requests before it waits for an HTTP response to the first request. The default thread count setting is 5 for a crawl and 10 for an audit (if using separate requestors). In some environments, you may need to specify a lower number to avoid application or server failure. For more information, see "Scan settings: Requestor" on page 369.
- If, for any reason, you do not want OpenText DAST to crawl and attack certain directories, you must specify those directories using the Excluded URLs feature of OpenText DAST settings (see "Scan settings: Session Exclusions" on page 372). You can also exclude specific file types and MIME types.
- By default, OpenText DAST is configured to ignore many binary files (images, documents, and so on) 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 audit speed. If proprietary documents are in use, determine the file extensions of the documents and exclude them within OpenText DAST's default settings. If, during a crawl, OpenText DAST becomes extremely slow or stops, it may be because it attempted to download a binary document.
- For form submission, OpenText DAST submits data extracted from a prepackaged file. If you require specific values (such as user names and passwords), you must create a file with the Web Form Editor tool and identify that file to OpenText DAST. For more information, see the Web Form Editor chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
- Unless you are conducting an interactive scan, turn off your CAPTCHA software. CAPTCHA is designed to prevent automation on web applications and can interfere with an automated scan of your web application.
- OpenText DAST tests for certain vulnerabilities by attempting to upload files to your server. If your server allows this, OpenText DAST will record this susceptibility in its scan report and attempt to delete the file. Sometimes, however, the server prevents file deletion. For this reason, search for and delete files with names that start with "CreatedByHP" as a routine part of your post-scan maintenance.

See also

"OpenText DAST overview" on page 27

"Quick start " below

Quick start

This topic provides information to help you get started with OpenText DAST. It includes links to more detailed information.

Update SecureBase

To ensure that you have up-to-date information about the OpenText DAST catalog of vulnerabilities, use the following procedure to update your vulnerabilities database.

1. Start OpenText DAST.
2. On the **Start Page**, click **Start Smart Update**.
   The Smart Update window opens and lists available updates.
3. Click **Update**.

Note: Update the product each time you use it. You can select an application setting that runs Smart Update each time you start the program. For more information, see "Application settings: Smart Update" on page 454.

For more information, including instructions for updating OpenText DAST that is offline, see "SmartUpdate" on page 279.

Prepare your system for audit

Before performing an audit, be aware of the potential impact on your website, and what you can do to prepare for a successful audit. For more information, see "Preparing your system for audit" on page 32.

Start a scan

After you update your database, you are ready to determine your web application's security vulnerabilities.

On the OpenText DAST **Start Page**, click one of the following selections:

- **Start a Guided Scan** (see "Guided Scan overview " on page 95)
- **Start a Basic Scan** (see "Running a Basic Scan (website scan)" on page 171)
- **Start an API Scan** (see "Using the API Scan Wizard" on page 137)

See also

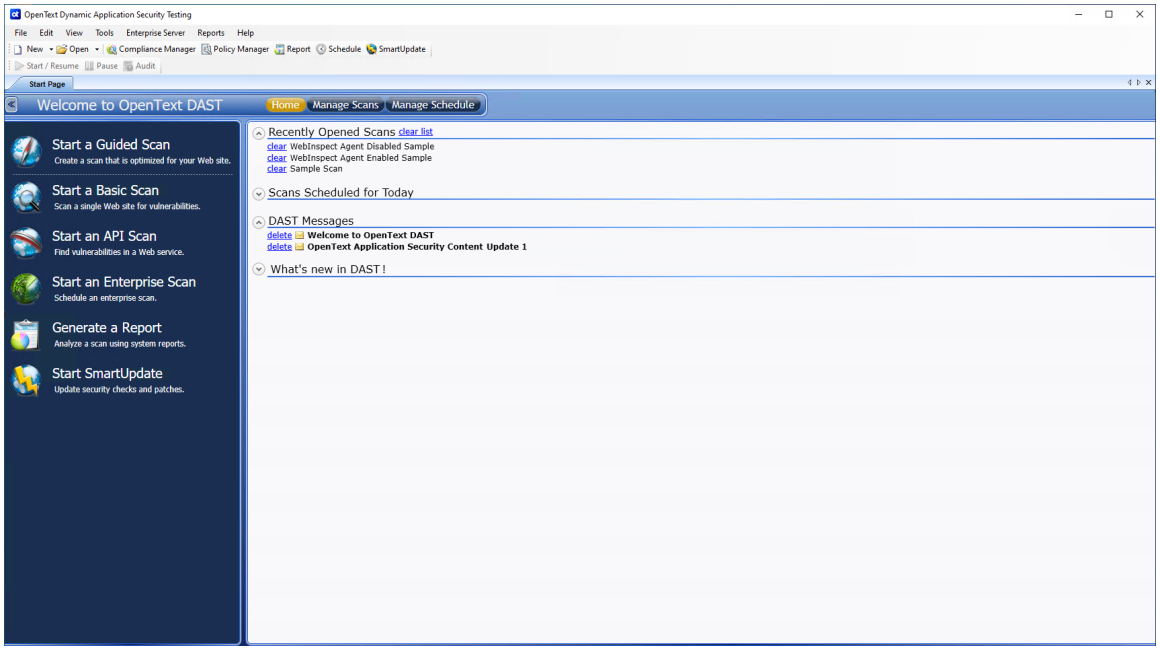
"Preparing your system for audit" on page 32

"OpenText DAST user interface" on page 36

Chapter 3: OpenText DAST user interface

When you first start OpenText DAST, the application displays the **Start Page**.

Start Page Image



The Activity Panel

The left pane (the Activity Panel) displays hyperlinks to the following major functions:

- Start a Guided Scan (see "Guided Scan overview " on page 95)
- Start a Basic Scan (see "Running a Basic Scan (website scan)" on page 171)
- Start an API Scan (see "Using the API Scan Wizard" on page 137)
- Generate a Report (see "Generating a report" on page 260)
- Start SmartUpdate (see "SmartUpdate" on page 279)

Closing the Activity Panel

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

The Button Bar

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

The choices are described in the following table.

Button	Displayed List
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 OpenText server. If you hover the pointer over a scan name, OpenText DAST displays summary information about the scan. If you click the scan name, OpenText DAST opens the scan on a separate tab.
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 a 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. For more information, see "Manage Scans" on page 210.
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. For more information, see "Managing scheduled scans" on page 213.

Panes Associated with a Scan

Each time you open or conduct a scan, OpenText DAST opens a tab labeled with the name or description of the target site.

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 tab bar. Click the **X** to close the selected tab.

See also

"About the menu bar " on the next page
"Toolbars" on page 43
"Start Page " below
"Navigation pane" on page 46
"Summary pane" on page 87
"Information pane" on page 57

Start Page

The left-hand pane of the **Start Page** contains a list of activities related to the vulnerability scan of your website, API, or web service:

- Start a Guided Scan (see "Guided Scan overview " on page 95)
- Start a Basic Scan (see "Running a Basic Scan (website scan)" on page 171)
- Start an API Scan (see "Using the API Scan Wizard" on page 137)
- Generate a Report (see "Generating a report" on page 260)
- Start SmartUpdate (see "SmartUpdate" on page 279)

The contents of the right-hand pane are controlled by the buttons on the Button bar.



Home

When **Home** is selected (the default), OpenText DAST displays a list of:

- Recently opened scans.
  If you hover the pointer over a scan name, OpenText DAST displays summary information about the scan. If you click the scan name, OpenText DAST opens the scan on a separate tab.
- Scans scheduled to be conducted today
- Recently generated reports
- Messages downloaded from the OpenText server

Manage Scans

When **Manage Scans** is selected, OpenText DAST 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 the SQL Server database, if configured), or both. For more information, see "Manage Scans" on page 210.

Manage Schedule

When **Manage Schedule** is selected, OpenText DAST displays a list of scheduled scans. You can add a scan to the schedule, edit or delete a scheduled scan, or start the scan manually. For more information, see "Managing scheduled scans" on page 213.

See also

"OpenText DAST user interface" on page 36

About the menu bar

The menu bar options are:

- "File menu" below
- "Edit menu " on the next page
- "View menu " on page 41
- "Tools menu " on page 41
- "Scan menu " on page 41
- "Reports menu " on page 42
- "Help menu" on page 43

File menu

The **File** menu commands are described in the following table.

Command	Description
New	Enables you to select Guided Scan, Basic Scan, or API Scan, and then launches the associated Scan Wizard, which steps you through the process of starting a scan.
Open	Enables you to open either a scan or a generated report.
Schedule	Opens the Manage Scheduled Scans window, which enables you to add, edit, or delete a scheduled scan.
Import Scan	Enables you to import a scan file.
Export	This command is available only when a tab containing a scan is selected. You may:

Command	Description
	• Export a scan • Export scan details • Export a scan to Application Security Center • Export protection rules to a web application firewall (WAF)
Close Tab	When multiple tabs are open, closes the selected tab. **Tip:** You can right-click any open tab and use the context-menu options as follows: • **Close** - Close the tab that you clicked • **Close All But This** - Close all tabs except the tab that you clicked • **Close All** - Close all tabs Likewise, you can use a middle-click or `CTRL`+`F4` to close a single tab. If you are closing the tab for a retest scan, you may be prompted about keeping the scan. For more information, see "Retesting vulnerabilities" on page 226.
Exit	Closes the OpenText DAST program.

Edit menu

The **Edit** menu commands are described in the following table.

Command	Description
Default Scan Settings	Displays the Default Settings window, allowing you to select or modify options used for scanning.
Current Scan Settings	Displays a settings window that enables 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 enables you to add, edit, or delete settings files.
Application	Displays the Application Settings window, allowing you to select or

Command	Description
Settings	modify options controlling the operation of the OpenText DAST application. For more information, see the "Application settings" on page 431.
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 **View** menu commands are described in the following table.

Command	Description
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	Enables you to select which toolbars should be displayed. For more information, see "Toolbars" on page 43.

Tools menu

The **Tools** menu contains commands to launch the tool applications.

Scan menu

The **Scan** menu appears on the menu bar only when a tab containing a scan has focus. Scan menu commands are described in the following table.

Command	Description
Start/Resume	Starts or resumes a scan after you paused the process.
Pause	Halts a crawl or audit. Click **Resume** to continue the scan.

Command	Description
Audit	Assesses the crawled site for vulnerabilities. Use the command after completing a crawl or exiting Step Mode.
Rescan	Launches the Scan Wizard prepopulated with settings last used for the selected scan. The Rescan drop-down menu enables you to: • Scan Again (see "Rescanning a site" on page 231) • Retest Vulnerabilities (see "Retesting vulnerabilities" on page 226) • Reuse Incremental (see "Incremental scanning" on page 233) • Reuse Remediation (see "Reusing scans" on page 232)
Compare	Compares the vulnerabilities revealed by two different scans of the same target. See "Comparing scans" on page 204.

Reports menu

The **Reports** menu commands are described in the following table.

Command	Description
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.

Help menu

The **Help** menu provides the commands described in this topic.

DAST Help

This command opens the Help file.

Support

This command displays instructions for contacting Customer Support.

Tutorials

This command opens the Fortify Unplugged YouTube™ channel for OpenText DAST.

DAST Community

This command opens the Fortify Community that offers discussions, tips, security blogs, news, and events.

About DAST

This command displays information about the OpenText DAST application, including license information, allowed hosts, and attributes.

Toolbars

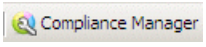
The OpenText DAST window contains two toolbars: Scan and Standard. You can display or hide either toolbar by selecting **Toolbars** from the **View** menu.

Buttons available on the Scan toolbar

Button	Function
	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**, OpenText DAST will attempt to resend those sessions.

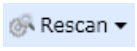
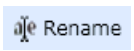
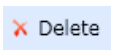
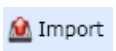
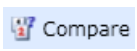
Button	Function
	Interrupts an ongoing scan. You can continue scanning by clicking the **Start/Resume** button.
	If you conduct a crawl-only scan or a Step Mode scan, you can afterwards click this button to conduct an audit. For more information, see "Running a manual scan" on page 197.
	This button appears only if you select a tab containing a scan. The Rescan drop-down menu enables you to: • Scan Again (see "Rescanning a site" on page 231) • Retest Vulnerabilities (see "Retesting vulnerabilities" on page 226) • Reuse Incremental (see "Incremental scanning" on page 233) • Reuse Remediation (see "Reusing scans" on page 232) For more information, see "Retesting and rescanning" on page 226.
	This button appears only if you select a tab containing a scan. It enables you to compare the vulnerabilities revealed by two different scans of the same target. For more information, see "Comparing scans" on page 204.

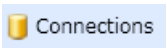
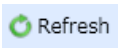
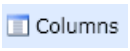
Buttons available on the Standard toolbar

Button	Function
	Enables you to select Guided Scan, Basic Scan, or API Scan, and then launches the associated Scan Wizard, which steps you through the process of starting a scan.
	Enables you to open a scan or a report.
	Starts the Compliance Manager.
	Starts the Policy Manager.

Button	Function
	Starts the Report Generator.
	Enables you to schedule a scan to occur on a specific time and date. For more information, see "Schedule a scan" on page 212.
	Contacts the central OpenText database to determine if updates are available for your system and, if updates exist, enables you to install them. For more information, see "SmartUpdate" on page 279.

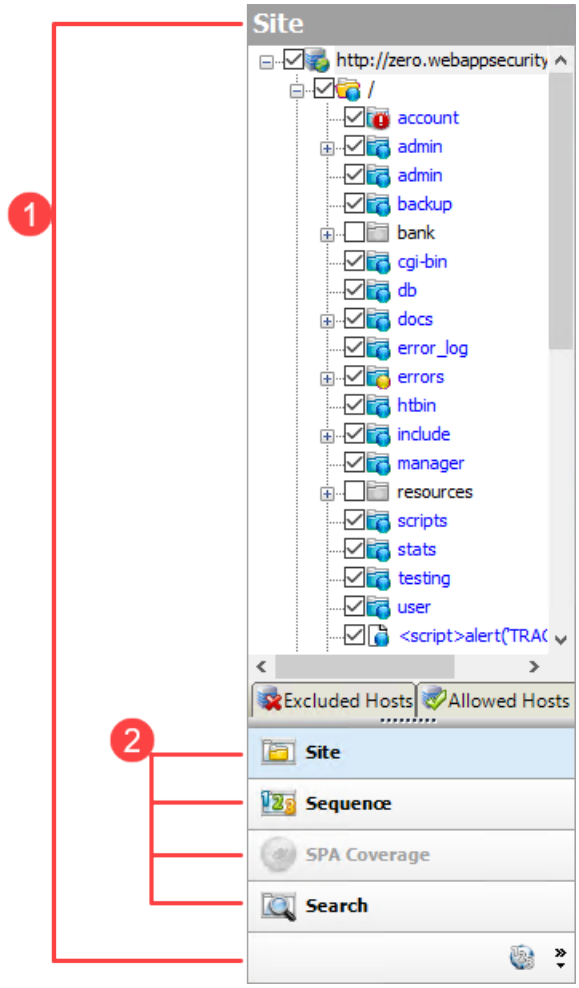
Buttons available on the Manage Scans toolbar

Button	Function
	To open scans, select one or more scans and click **Open** (or simply double-click an entry in the list). OpenText DAST loads the scan data and displays each scan on a separate tab.
	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 "Retesting and rescanning" on page 226.
	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 Application Security, or to export protection rules to a web application firewall (WAF), click the drop-down arrow on **Export**.
	To compare scans, select two scans (using Ctrl + click) and

Button	Function
	click **Compare**. For more information, see "Comparing scans" on page 204.
	By default, OpenText DAST 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.

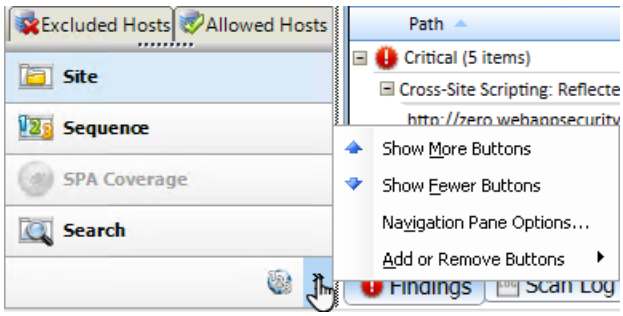
Navigation pane

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



Item	Description
1	Navigation pane
2	Buttons for changing the view

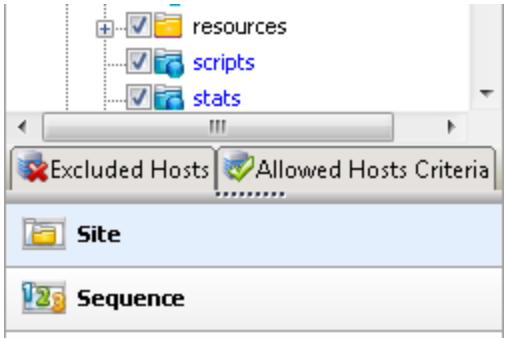
If all buttons are not displayed, click the drop-down arrow at the bottom of the button list and select **Show More Buttons**.



Site view

OpenText DAST displays in the navigation pane only the hierarchical structure of the website or web service, plus those sessions in which a vulnerability was discovered. During the crawl of the site, OpenText DAST 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 and then audited), you can exclude a session from the audit by clearing its associated check box before the audit begins.

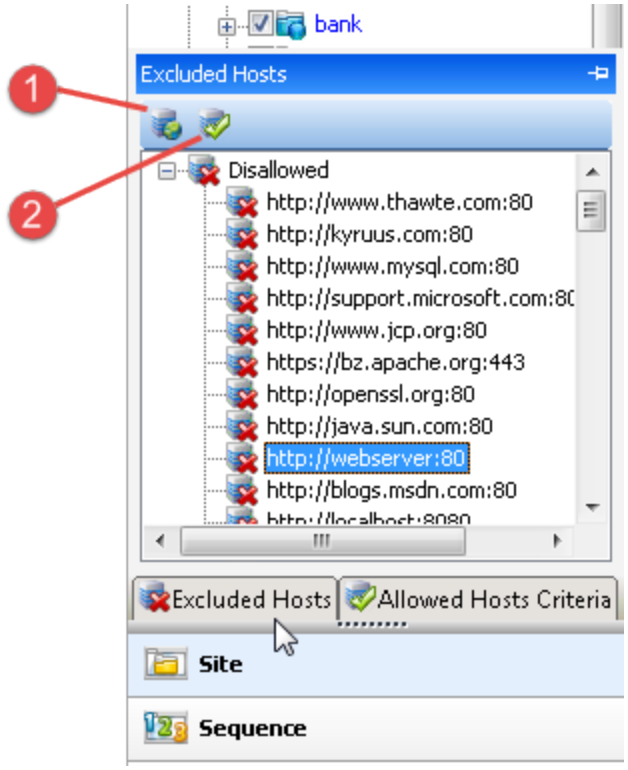
Site view also contains two pop-up tabs: **Excluded Hosts** and **Allowed Hosts Criteria**.



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).

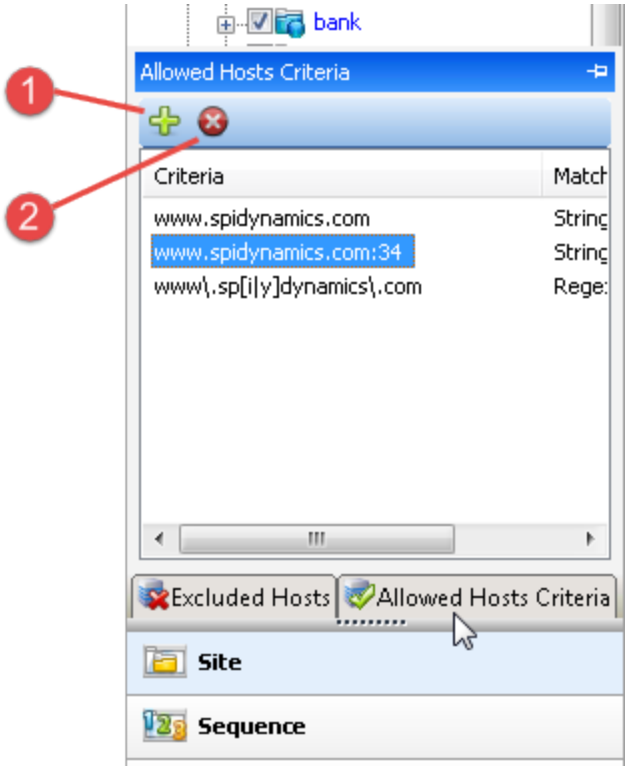
Using the **Excluded Hosts** tab, you can select an excluded host and click either **Add to scan** or **Add allowed host criteria**.



Item	Description
1	Add to scan – Adding a host to the scan creates a node in the site tree representing the host root directory. OpenText DAST will scan that session.
2	Add to Allowed Host Criteria – Adding a host to the allowed host criteria adds the URL to the list of allowed hosts in the Current Scan Settings. OpenText DAST will include in the scan any subsequent links to that host. However, if you add a host to the allowed host criteria after OpenText DAST 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 OpenText DAST scan settings (under Allowed Hosts). If you click either **Delete** or **Add allowed host criteria**, OpenText DAST opens the Current Settings dialog box, where you can add, edit, or delete allowed host criteria (a literal URL or a regular expression representing a URL).



Item	Description
1	Add Allowed Host Criteria – If you add an entry, OpenText DAST will include in the scan any subsequent links it encounters to hosts that match the criteria. However, if you specify a host after OpenText DAST has already scanned the only resource containing a link to that host, the added host will not be scanned.
2	Delete – If you delete an entry from the allowed host list, the scan will still include any resources that OpenText DAST 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 OpenText DAST has already scanned the only resource containing a link to the added host, the added host will not be scanned.

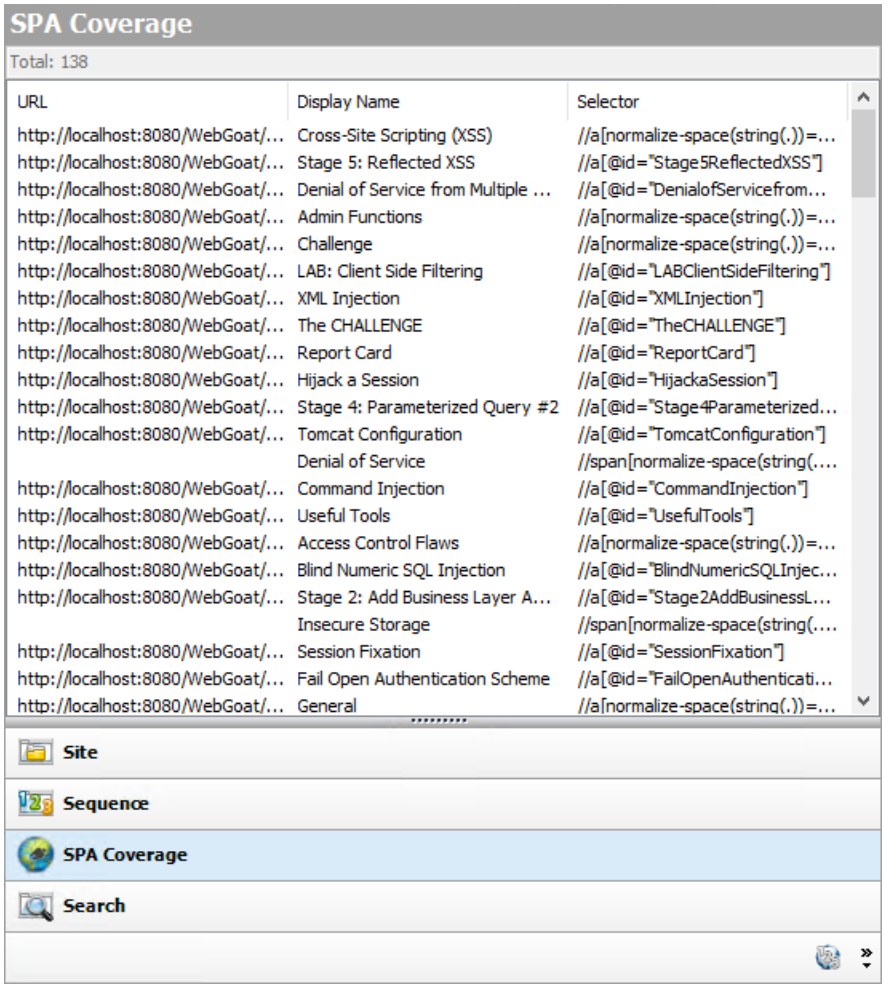
Sequence View

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

Note: In both Site view and Sequence view, blue text denotes a directory or file that was "guessed" by OpenText DAST, rather than a resource that was discovered through a link. For example, OpenText DAST always submits the request "GET /backup/ HTTP/1.1" in an attempt to discover if the target website contains a directory named "backup."

SPA coverage

The SPA Coverage view is available only if SPA support is enabled for a scan. This view displays the elements in the page that the crawler interacted with during the crawl.



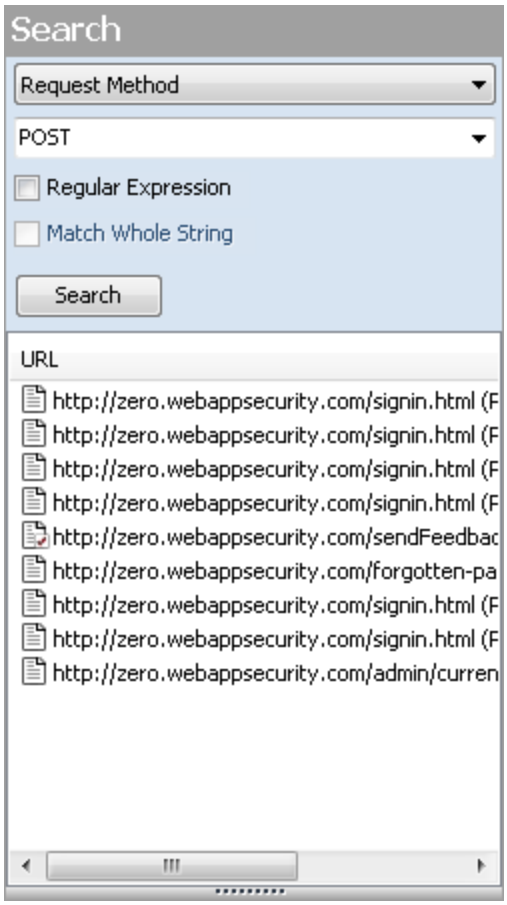
The SPA Coverage view lists the URLs where the elements were discovered, along with the following additional information:

- **Display Name** – The visible text, symbol, link, HTML tag name, or other UI information related to the element discovered.
- **Selector** – The XPath location of the element in the page. This is used to find and perform operations on the element.

For more information, see "About single-page application scans" on page 201.

Search view

The Search view enables you to search across all sessions for various HTTP message components. For example, if you select **Request Method** from the drop-down list and specify **POST** as the search string, OpenText DAST lists every session whose HTTP request uses the POST method.



To use the Search view:

1. In the navigation pane, click **Search** (at the bottom of the pane).

   If all buttons are not displayed, click the **Configure Buttons** drop-down at the bottom of the button list and select **Show More Buttons**.
2. From the top-most list, select an area to search.
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. For more information, see "Regular expressions" on page 310.
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.

Note: This option is not available for certain search targets.

6. Click **Search**.

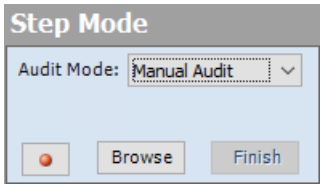
Step Mode view

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**.

   The selected 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 OpenText DAST and click **Finish**.

   The new sessions are added to the navigation pane.
7. If you selected **Manual Audit** in step 3, click Audit. OpenText DAST 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 navigation pane.

Icons used in the Navigation pane

Icon	Description
	Server/host: Represents the top level of your site's tree structure.
	Blue folder: A folder discovered by "guessing" and not by crawling.

Icons used in the Navigation pane , continued

Icon	Description
	Yellow folder: A folder whose contents are available over your website.
	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.
	DOM event.

Icons superimposed on a folder or file indicate a discovered vulnerability

Icon	Description
	A red dot with an 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.

Navigation pane shortcut menu

If you right-click an item in the navigation pane while using the Site or Sequence view, a shortcut menu presents the following options:

- **Expand Children*** - (Site View only) Expands branching nodes in the site tree.
- **Collapse Children*** - (Site View only) Contracts branching nodes into the superior node.
- **Check All*** - (Site View only) Marks the check box the parent node and all children.
- **Uncheck All*** - (Site View only) Removes the check mark from the parent node and 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 to the Windows clipboard.
- **View in Browser** - Renders the HTTP response in a 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, OpenText DAST 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** - Enables you to add locations discovered by means other than an OpenText DAST scan (manual inspection, other tools, etc) for information purposes. You can then add any discovered vulnerabilities to those locations so that a more complete picture of the site is archived for analysis.
  - **Page** - A distinct URL (resource).
  - **Directory** - A folder containing a collection of pages.

    Choosing either **Page** or **Directory** invokes a dialog box that enables you to name the directory or page and edit the HTTP request and response.
  - **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 Add Variation dialog box, 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 Edit Vulnerabilities dialog box, allowing you to edit the variation attributes, specify *Post* or *Query*, and edit the HTTP request and response. For more information, see "Editing vulnerabilities" on page 251.

- **Edit Vulnerabilities** - Enables you to edit a location that was added manually or edit a vulnerability. For more information, see "Editing vulnerabilities" on page 251.
- **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 sessions" on page 258 for details.

- **Mark as** - Flags the vulnerability as a false positive and enables you to add a description. The vulnerability is removed from the list. You can view a list of all false positive and ignored vulnerabilities by selecting **Suppressed Findings** in the Scan Info panel.

  Note: You can change a suppressed finding back to a vulnerability. See "Suppressed findings" on page 67 for details.

- **Send to** - Enables you convert the selected vulnerability to a defect and assign it to OpenText Application Lifecycle Management (ALM), using the profile specified in the OpenText DAST 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** - Enables 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.*

See also

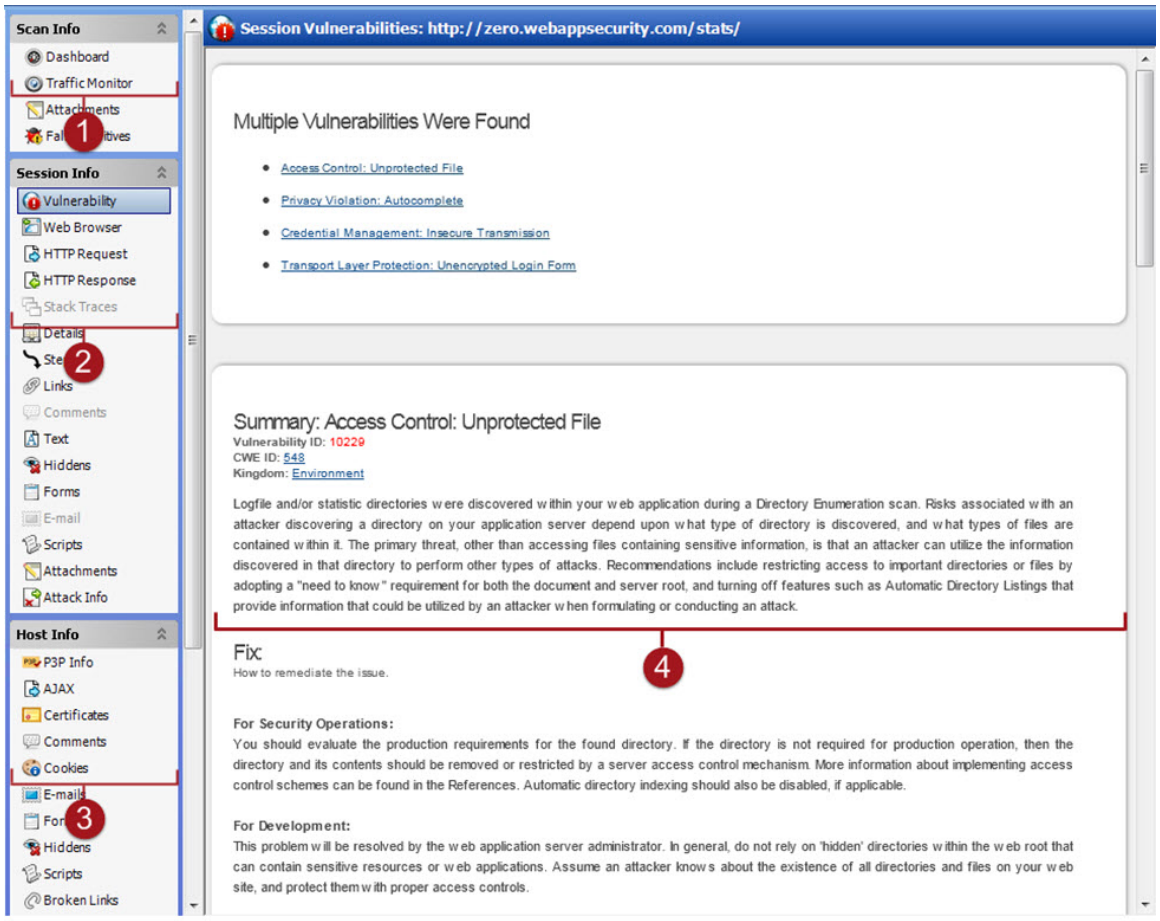
"OpenText DAST user interface" on page 36

"Search view" on page 244

"Inspecting the results" on page 239

Information pane

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



Item	Description
1	Scan Info panel (See "Scan Info panel" on the next page)
2	Session Info panel (See "Session Info panel" on page 69)
3	Host Info panel (See "Host Info panel" on page 79)
4	Information display area

Select the type of information to display by clicking on an item in one of these 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.

See also

"Summary pane" on page 87

"OpenText DAST user interface" on page 36

"Navigation pane" on page 46

"Scan Info panel" below

"Session Info panel" on page 69

"Host Info panel" on page 79

Scan Info panel

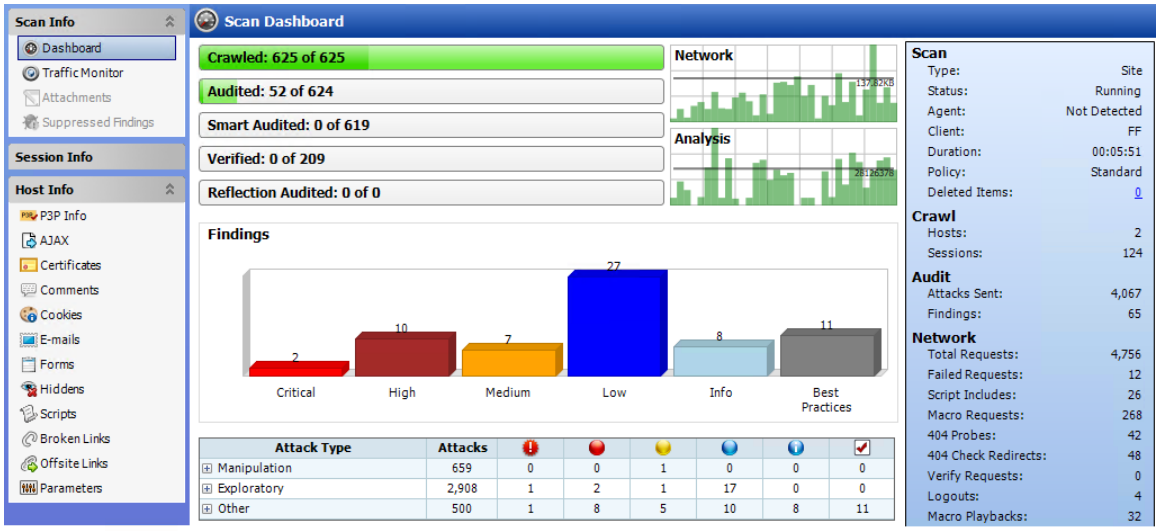
The **Scan Info** panel has the following choices:

- Dashboard
- Traffic Monitor
- Attachments
- Suppressed Findings

Dashboard

The **Dashboard** selection displays a real-time summary of the scan results and a graphic representation of the scan progress. This section is displayed only if you select this option from the Default or Current settings. For additional information, see "Dashboard" on page 60.

Dashboard Image



Traffic Monitor

OpenText DAST normally displays in the navigation pane only the hierarchical structure of the website or web service, plus those sessions in which a vulnerability was discovered. The Traffic Monitor or Traffic Viewer enables you to display and review every HTTP request sent by OpenText DAST and the associated HTTP response received from the web server.

The Traffic Monitor or Traffic Viewer is available only if Traffic Monitor Logging was enabled prior to conducting the scan.

For more information, see "Traffic Monitor (Traffic Viewer)" on page 237.

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

To edit an attachment, select the attachment and click **Edit**.

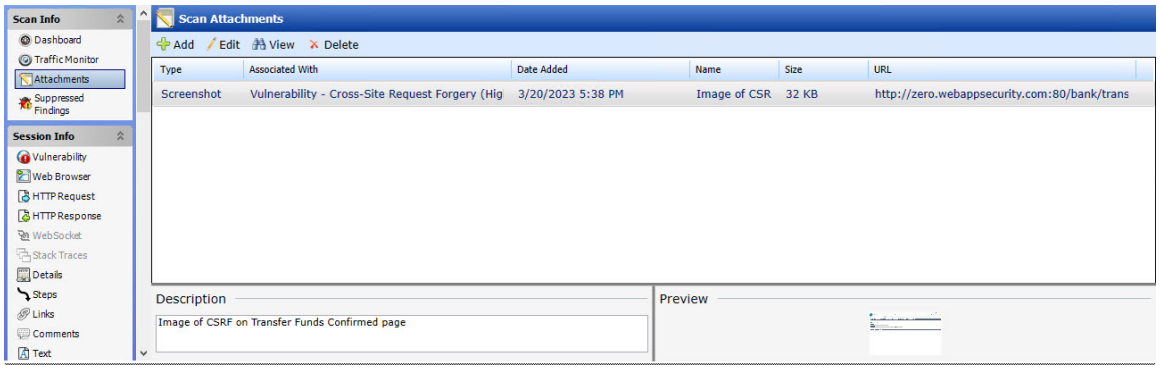
To create attachments in other areas of the OpenText DAST user interface, you can do one of the following:

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

OpenText DAST automatically adds a note to the session whenever you send a defect to OpenText Application Lifecycle Management (ALM).

For more information, see "Attachments - Scan Info" on page 66.

Attachments Image

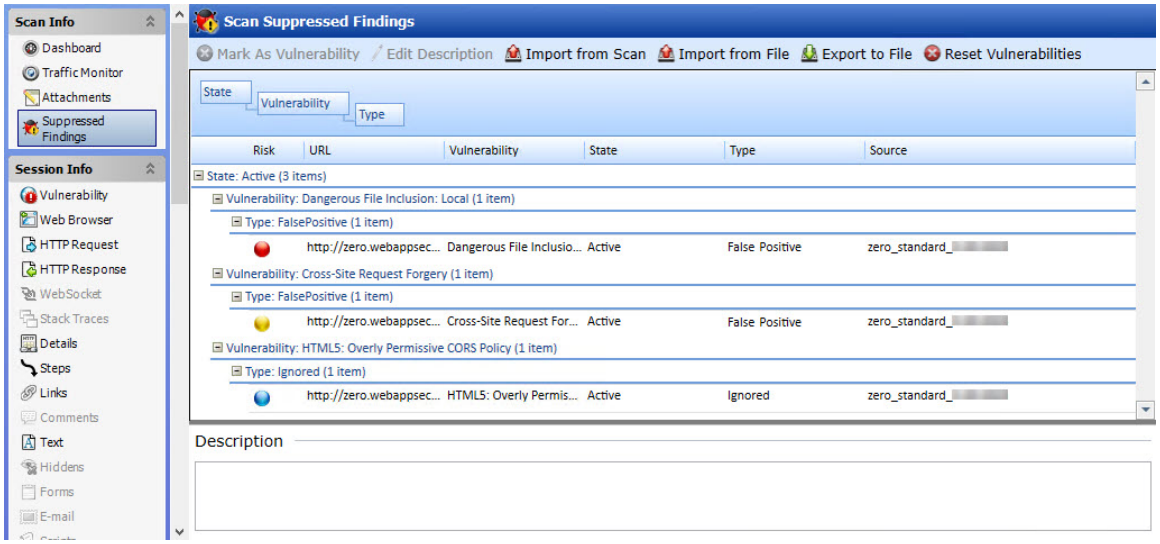


Suppressed Findings

This feature lists all URLs that OpenText DAST originally flagged as containing a vulnerability, but which a user later determined were false positives or chose to ignore.

For more information, see "Suppressed findings" on page 67.

Suppressed Findings Image



See also

"Session Info panel" on page 69

"Host Info panel" on page 79

"OpenText DAST user interface" on page 36

"Dashboard" below

"Traffic Monitor (Traffic Viewer)" on page 237

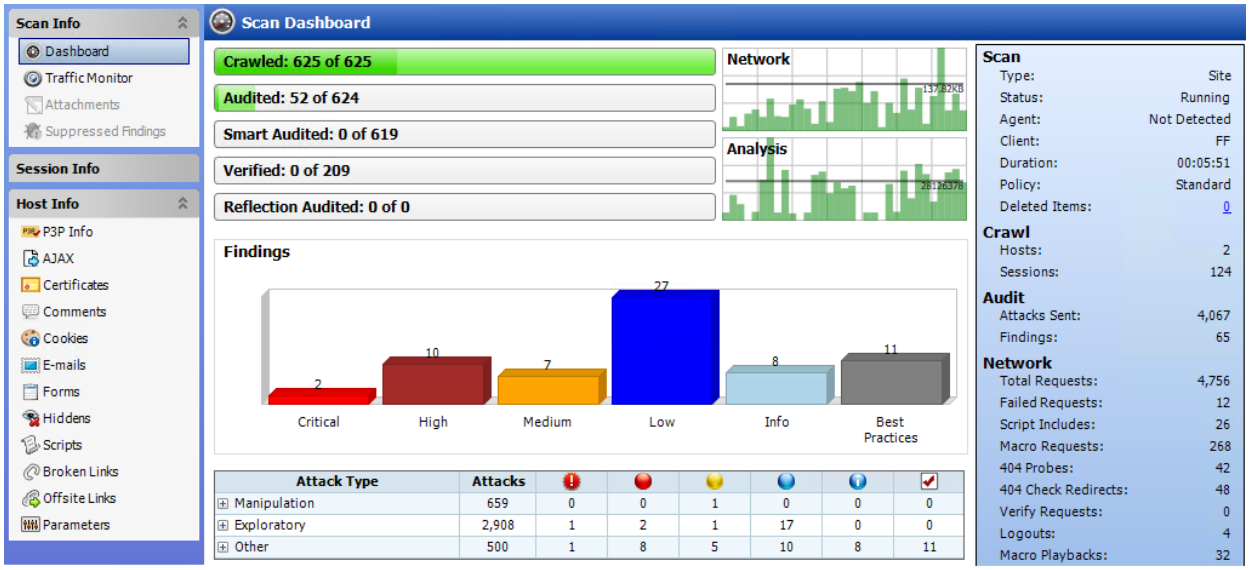
"Attachments - Scan Info" on page 66

Dashboard

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

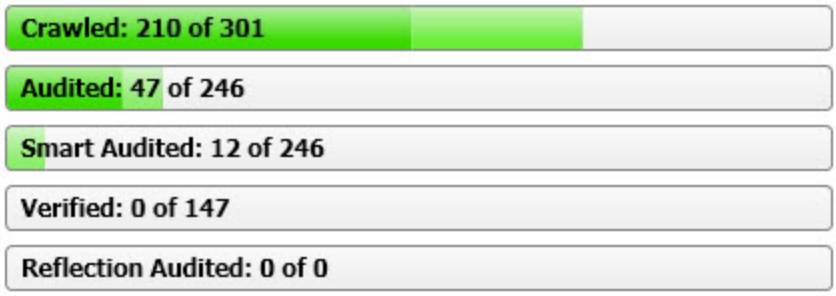
Dashboard Image

The following image displays the Scan Dashboard with a scan in progress.



Progress bars

Each bar represents the progress being made through that scanning phase.



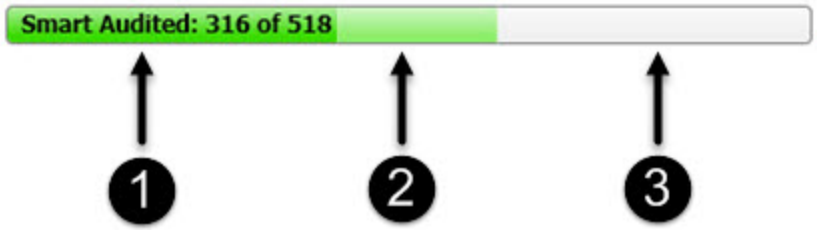
Progress bar descriptions

The following table describes the progress bars.

Progress Bar	Description
Crawled	Number of sessions crawled / total number of sessions to crawl.
Audited	Number of sessions audited / total number of sessions to audit. The total number includes all checks except those pertaining to server type, which are handled by smart audit.
Smart Audited	Number of sessions audited using smart audit / total number of sessions for smart audit.

Progress Bar	Description
	For smart audit, OpenText DAST detects the type of server on which the Web application is hosted. OpenText DAST runs checks that are specific to the server type and avoids checks that are not valid for the server type.
Verified	Number of persistent XSS vulnerable sessions verified / total number of persistent XSS vulnerable sessions to verify. When persistent XSS auditing is enabled, OpenText DAST sends a second request to all vulnerable sessions and examines all responses for probes that OpenText DAST previously made. When probes are located, OpenText DAST will record links between those pages internally.
Reflection Audited	Number of persistent XSS vulnerable linked sessions audited / total number of persistent XSS vulnerable linked sessions to audit. When persistent XSS auditing is enabled, this represents the work required for auditing the linked sessions found in the verification step for persistent XSS.

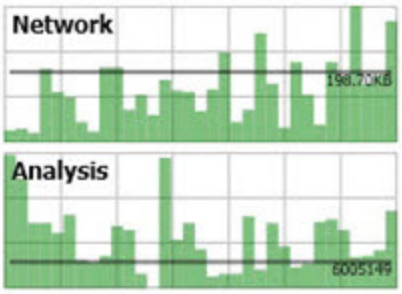
Progress bar colors



1. Dark green indicates sessions that have been processed.
2. Light green indicates excluded, aborted, or rejected sessions (sessions that were considered for processing, but were skipped due to settings or other reasons).
3. Light gray indicates the unprocessed sessions.

Activity meters

OpenText DAST polls information about the activity occurring in the scan and displays the data in activity meters. The data presents a real-time snapshot of the scan activity. This information can help you to determine whether the scan is stalled or actively running.



Activity meter descriptions

The following table describes the activity meters.

Meter	Description
Network	The amount of data being sent and received by OpenText DAST. The chart shows this data as B, KB, or MB sent/received over the last one second.
Analysis	The amount of work being done per second by OpenText DAST in processing all threads.

Findings graphics

The following table describes the Findings bar graph and grid.

Graphic	Description
Findings Graph	Total number of issues identified for the scan per severity level.
Attack Stats Grid	Number of attacks made and issues found, categorized by attack type and audit engine.

Statistics panel - Scan

The following table describes the Scan section of the statistics panel.

Item	Description
Type	Type of scan: Site, Service, or Site Retest.
Scan Status	Status: Running, Paused, Interrupted, or Complete.
Agent	Refers to the OpenText DAST Agent and states either Detected or Not Detected. For certain checks (such as SQL injection, command

Item	Description
	execution, and cross-site scripting), OpenText DAST Agent intercepts OpenText DAST HTTP requests and conducts runtime analysis on the target module. If this analysis confirms that a vulnerability exists, OpenText DAST Agent appends the stack trace to the HTTP response. Developers can analyze this stack trace to investigate areas that require remediation.
Client	The rendering engine specified for the scan. Options are: • IE (Internet Explorer) • FF (Firefox) • iPhone • iPad • Android • Windows Phone • Windows RT
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. If the scan used multiple policies, the number of policies selected is shown as a hyperlink. Click the link to view the selected policies.
Deleted Items	The number of sessions and vulnerabilities removed by the user from the scan. To remove a session, right-click a session in the Navigation pane and select **Remove Location** from the shortcut menu. For more information, see "Navigation pane" on page 46. To remove a vulnerability, right-click a vulnerability in the Summary pane and select **Ignore Vulnerability** from the shortcut menu. For more information, see "Summary pane" on page 87. To restore sessions or vulnerabilities that have been deleted: 1. On the Scan Dashboard, click the number associated with deleted items. The Recover Deleted Items window appears. 2. Select either **Vulnerabilities** or **Sessions** from the drop-down menu.

Item	Description
	3. Select one or more items. 4. Click **Recover**.

Statistics panel - Crawl

The following table describes the Crawl section of the statistics panel.

Item	Description
Hosts	Number of hosts included in the scan.
Sessions	Total number of sessions (excluding AJAX requests, script and script frame includes, and WSDL includes).

Statistics panel - Audit

The following table describes the Audit section of the statistics panel.

Item	Description
Attacks Sent	Total number of attacks sent.
Issues	Total number of issues found (all vulnerabilities, as well as best practices).

Statistics panel - Network

The following table describes the Network section of the statistics panel.

Item	Description
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 file not found probes made to determine file not found status.

Item	Description
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.
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.

See also

"Scan Info panel" on page 58

"Session Info panel" on page 69

"Host Info panel" on page 79

Attachments - Scan Info

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). For more information, see "Information pane" on page 57.

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 areas of the OpenText DAST user interface, do one of the following:

- Right-click a session in the navigation pane and select **Attachments** from the shortcut menu. For more information, see "Navigation pane" on page 46.
- Right-click a URL on the **Findings** tab of the summary pane and select **Attachments** from the shortcut menu. For more information, see "Findings tab" on page 88.

OpenText DAST automatically adds a note to the session whenever you send a defect to OpenText Application Lifecycle Management (ALM).

See also

"Scan Info panel" on page 58

Suppressed findings

This feature lists all URLs that OpenText DAST originally flagged as containing a vulnerability, but which have been marked as "False Positive" or "Ignore" by a user.

Understanding suppressed findings

OpenText DAST allows the following types of suppressed findings:

- **False Positive** - A finding that upon further investigation by a developer is determined not to be a vulnerable URL, operation, or parameter.
- **Ignored** - A finding that a security lead or developer has chosen to ignore. Generally, these should be low-level or informational findings that carry little risk of exploitation, or have mitigation that is outside the scope of testing.

  For example, you perform a scan on a developer machine that is not configured the same way as your server, such as using a self-signed certificate that isn't secure. In production, however, the server uses a real certificate that is secure. During the scan, OpenText DAST flags this vulnerability, but it is not an issue that needs to be fixed, so you ignore it.

Importing suppressed findings

You can import from a previous scan a list of vulnerabilities that were analyzed as being false positive or were ignored. OpenText DAST then correlates these suppressed findings from the previous scan with vulnerabilities detected in the current scan and flags the new occurrences as false positive or ignored.

For example, suppose a cross-site scripting vulnerability was detected in Scan No. 1 at URL http://www.mysite.com/foo/bar and, after further analysis, a developer 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 OpenText DAST automatically changes that vulnerability to a false positive.

Inactive / Active Suppressed Findings lists

Imported suppressed findings are loaded first into a list labeled "Inactive Suppressed Findings." If a suppressed finding in that list is matched with a vulnerability in the current scan, the item is moved from the Inactive Suppressed Findings list to the Active Suppressed Findings list. Unmatched items remain in the Inactive Suppressed Findings list.

Loading suppressed findings during scan configuration

While configuring a scan in the Scan Wizard, you can choose to load suppressed findings from a specific scan or file. During the scan, OpenText DAST correlates the suppressed findings as they are encountered. You can also see on the scan dashboard the false positives matched while the scan is running.

Working with suppressed findings

1. Select **Suppressed Findings** 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. For false positive items, click a URL to view a comment (at the bottom of the Information pane) that may have been entered when the user marked the vulnerability as **False Positive**.
4. Continue according to the following table:

To...	Then...
Import suppressed findings from other scans	Continue with "Selecting a scan to import suppressed findings" on page 224.
Import suppressed findings from a JSON file	a. Click **Import from File**. A standard Windows file selection dialog box opens. b. Select the file to import, and then click **Open**.
Export suppressed findings to a JSON file	a. Click **Export to File**. A standard Windows save as dialog box opens. **Note:** The default directory for suppressed findings exported to a file is `<directory>:\ProgramData\HP\HP WebInspect\Settings\SuppressedFindings`.

To...	Then...
	b. In the **File name** box, type a name for the suppressed findings file. c. Click **Save**.
Change a suppressed finding back to a vulnerability	a. Select an item from the Active **Suppressed Findings** list and click **Mark as Vulnerability**. The Mark As Vulnerability dialog box opens. b. Continue with "Mark as vulnerability" on page 255.
Change all suppressed findings back to vulnerabilities	a. Click **Reset Vulnerabilities**. A confirmation dialog box opens. b. Click **Yes**. The suppressed findings are added back to the Findings tab.
Remove an item from the Inactive Suppressed Findings list	Select the item and click **Remove From Inactive**.
Edit the description for a false positive	a. Select the item and click **Edit Description**. The Edit False Positive Description dialog box opens. b. Edit the description, and then click **OK**.

See also

For information on how to designate a vulnerability as a false positive, see "Navigation pane shortcut menu" on page 55 or "Findings tab" on page 88.

For more information on the OpenText DAST window, see "OpenText DAST user interface" on page 36.

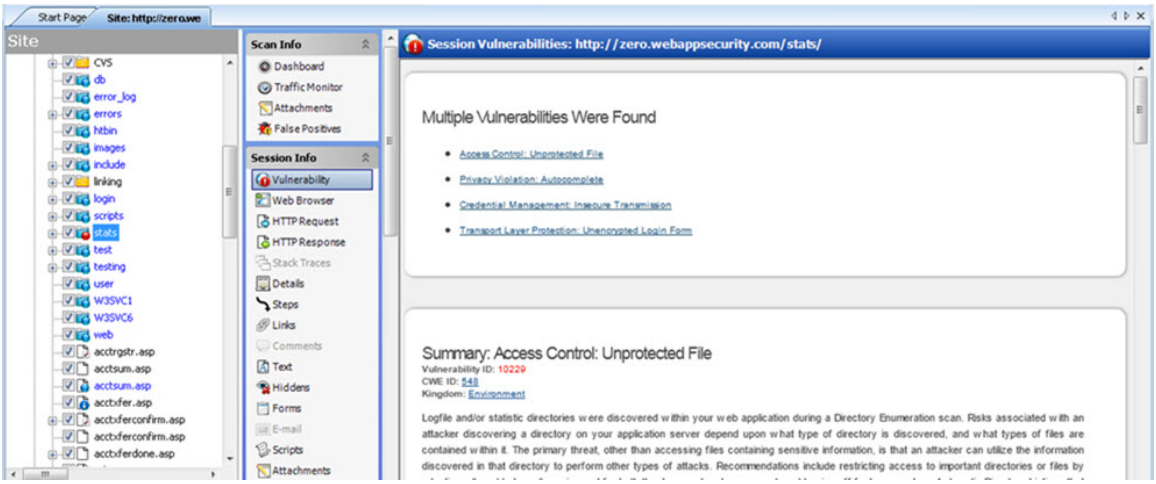
Session Info panel

OpenText DAST 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 following example scan, OpenText DAST sent the HTTP request GET /stats/stats.html HTTP/1.1.

To see the vulnerability:

1. Select **Stats.html** in the navigation pane.
2. In the Session Info panel, click **Vulnerability**.



Options available

The following table lists the options available in the **Session Info** panel. Some options appear only for specific scans (Basic 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.

Option	Description
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 OpenText DAST to the server hosting the site you are scanning.
HTTP Response	Displays the server's raw HTTP response to OpenText DAST'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:  If you select a Flash (.swf) file, OpenText DAST displays HTML instead of binary data. This allows OpenText DAST to display links in

Option	Description
	a readable format.
Stack Traces	This feature is designed to support OpenText DAST Agent when it is installed and running on the target server. For certain checks (such as SQL injection, command execution, and cross-site scripting), OpenText DAST Agent intercepts OpenText DAST HTTP requests and conducts runtime analysis on the target module. If this analysis confirms that a vulnerability exists, OpenText DAST Agent appends the stack trace to the HTTP response. Developers can analyze this stack trace to investigate areas that requires remediation.
Details[1]	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 OpenText DAST.
Steps[1]	Displays the route taken by OpenText DAST 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.
Links[1]	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 that are referenced by links within the HTTP response for the selected session.
Comments[1]	Displays all comments (in HTML) embedded in the HTTP response.
Text[1]	Displays all text contained in the HTTP response for the session selected in the navigation pane.
Hiddens[1]	Displays the name attribute of each input element whose control type is "hidden."
Forms[1]	Displays the HTML interpreted by the browser to render forms.

Option	Description
E-mail[1]	Displays all email addresses included in the response.
Scripts[1]	Displays all client-side scripts embedded in the server's response.
Attachments	Displays all notes, flags, and screenshots associated with the selected object. To create an attachment, you can either: • Right-click a session (Basic or Guided 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 **Findings** tab of the summary pane and select **Attachments** from the shortcut menu, or • Select a session (Basic 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). OpenText DAST automatically adds a note to the session information whenever you send a defect to OpenText Application Lifecycle Management (ALM).
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**.
XML Request[2]	Displays the SOAP envelope embedded in the request (available when selecting an operation during a Web Service Scan).
XML Response[2]	Displays the SOAP envelope embedded in the response (available when selecting an operation 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] Basic or Guided 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** button before clicking **Find**.

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

See also

"OpenText DAST user interface" on page 36

"Host Info panel" on page 79

"Navigation pane" on page 46

"Scan Info panel" on page 58

"Summary pane" on page 87

"Regular expressions" on page 310

Vulnerability

This option displays the vulnerability information for the session selected in the navigation pane or for the vulnerability selected in the summary pane. It typically includes a description of the vulnerability, vulnerability ID, Common Weakness Enumeration (CWE) ID, Kingdom, implications (how this vulnerability may affect you), and instructions on how to fix the vulnerability.

Web browser

This option displays the server's response as rendered by a web browser for the session selected in the Navigation pane.

HTTP Request

This option displays the raw HTTP request (for the session selected in the navigation pane) sent by OpenText DAST to the server hosting the site you are scanning.

Highlighted text in the request

In the HTTP request, OpenText DAST highlights text as follows:

- Yellow highlighting indicates the GET, POST, or PUT status line and cookie headers.
- Red highlighting indicates the attack payload and a vulnerability, if detected.

HTTP Response

This option displays the server's raw HTTP response to OpenText DAST's request, for the session selected in the navigation pane.

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:



If you select a Flash (.swf) file, OpenText DAST displays HTML instead of binary data. This allows OpenText DAST to display links in a readable format.

Highlighted text in the response

In the HTTP response, OpenText DAST uses red highlighting to indicate a detected vulnerability.

Stack traces

This feature is designed to support OpenText DAST Agent when it is installed and running on the target server.

For certain checks (such as SQL injection, command execution, and cross-site scripting), OpenText DAST Agent intercepts OpenText DAST HTTP requests and conducts runtime analysis on the target module. If this analysis confirms that a vulnerability exists, OpenText DAST Agent appends the stack trace to the HTTP response. Developers can analyze this stack trace to investigate areas that require remediation.

Details

This option lists request and response details, such as the size of the response and the request method, for the session selected in the navigation pane.

Note that the Response section contains two entries for content type: returned and detected. **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 OpenText DAST.

Steps

This option displays the route taken by OpenText DAST 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.

Links

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 that are referenced by links within the HTTP response for the selected session.

If you double-click a listed link, focus in the navigation pane shifts to the referenced session. Alternatively, you can browse to the linked resource by viewing the session in the web browser (click **Web Browser**). For more information, see "Navigation pane" on page 46.

Comments - session info

This option displays all comments embedded in the HTTP response for the session selected in the navigation pane.

Developers sometimes leave critical information in comments that can be used to breach the security of a site. 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 compromise the security of your site.

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**.

You can copy comments to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

Text

This option displays all text contained in the HTTP response for the session selected in the navigation pane. For more information, see "Navigation pane" on page 46.

Hiddens: Session Info

OpenText DAST 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.

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**.

You can copy the HTML text to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

Forms: Session Info

OpenText DAST lists all HTML forms discovered for the session selected in the navigation pane.

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**.

You can copy forms to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

For more information on the OpenText DAST window, see "OpenText DAST user interface" on page 36.

E-mail

OpenText DAST lists all email addresses contained in the session selected from the navigation pane.

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**.

You can copy email addresses to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

Scripts - Session Info

OpenText DAST lists all scripts discovered in a session.

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**. For more information, see "Regular expressions" on page 310.

You can copy the script to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

For more information on the OpenText DAST window, see "OpenText DAST user interface" on page 36.

Attachments - Session Info

You can associate the following attachments with a session:

- Session Note
- Flag Session for Follow Up
- Vulnerability Note
- Vulnerability Screenshot

Note: You can also associate a note with a scan and view all attachments that have been added to the scan by selecting **Attachments** in the **Scan Info** panel.

The **Attachments** selection displays a list of all notes, flags, and screenshots that have been associated with the selected session.

Viewing an attachment

To view an attachment:

- Select the attachment and click **View** (or simply double-click the attachment).

Adding a session attachment

To add a session attachment:

1. Do one of the following to select a session:
   - On the **Findings** tab in the Summary pane, right-click a vulnerable URL. For more information, see "Findings tab" on page 88.
   - On the Navigation pane, right-click a session or URL. For more information, see "Navigation pane" on page 46.
2. On the shortcut menu, click **Attachments** and select an attachment type.

   Note: An alternative method is to select a session in the Navigation pane, click **Attachments** in the **Session Info** panel, and then select a command from the **Add** menu (in the information display area). For more information, see "Information pane" on page 57.
3. Enter a comment related to the type of attachment you selected.
4. Select the check box next to one or more vulnerabilities.
5. If you selected **Vulnerability Screenshot**:
   a. Enter a name for the screenshot in the **Name** box. Maximum length is 40 characters.
   b. Click the Browse button ... to locate the graphic file or, if you captured the image in memory, click **Copy from Clipboard**.
6. Click **OK**.

Editing an attachment

To edit an attachment:

1. Do one of the following:
   - To view all attachments that have been added to the scan, click **Attachments** in the **Scan Info** panel.
   - To view only those attachments that have been added to a specific session, click

Attachments in the **Session Info** panel and then click a session in the Navigation pane. You can also select a URL in the Summary pane.

2. Select an attachment and click **Edit**.
3. Modify the comments as required.

   Note: Screenshot attachments cannot be edited.

4. Click **OK**.

Tip: Add, Edit, View, and Delete functions are also available by right-clicking an attachment in the information display area and selecting an option from the shortcut menu.

Attack Info

For the session selected in the navigation pane, this option 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**.

Web service request

This option displays the web service schema and values embedded in the request (available when selecting an operation during a Web Service Scan). It is available only during a Web Service scan.

Web service response

This option displays the web service schema and values embedded in the response (available when selecting an operation during a Web Service scan). It is available only during a Web Service scan.

XML request

This option displays the associated XML schema embedded in the selected request (available when selecting the WSDL object during a web service scan).

XML response

This option displays the associated XML schema embedded in the response for the session selected in the navigation pane (available when selecting the WSDL object during a web service scan).

Host Info panel

When you click any item listed in this collapsible panel, OpenText DAST displays all instances of that item type that were discovered during a crawl or audit of the site (or host).

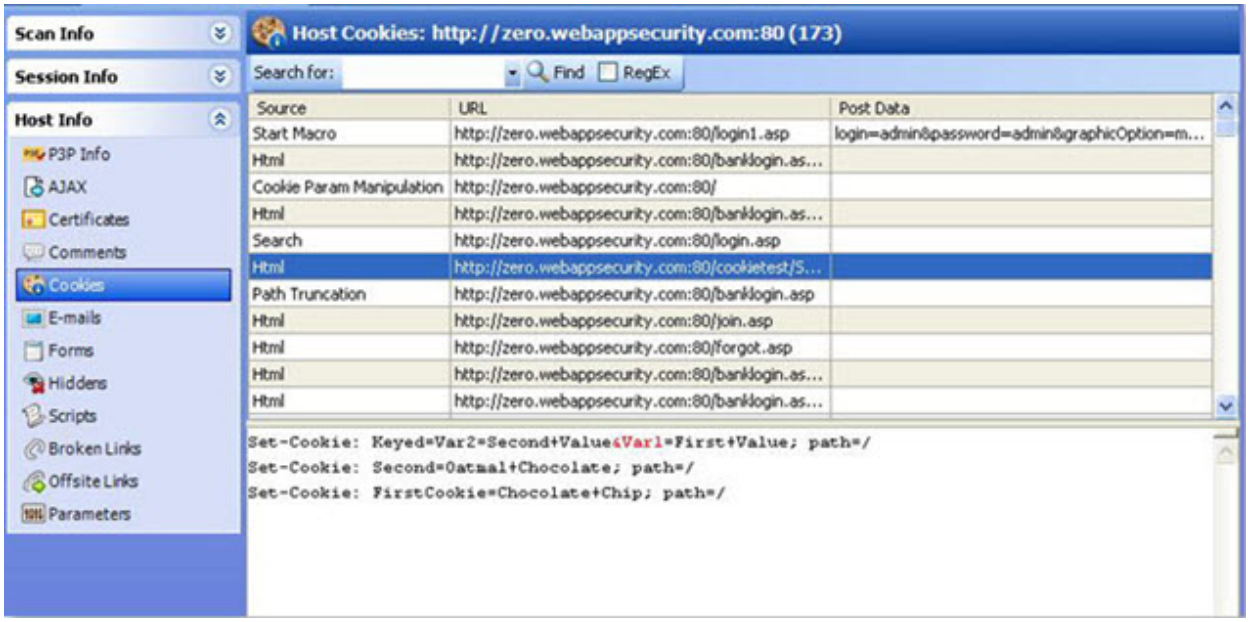
If you double-click an item, OpenText DAST highlights in the navigation pane the session that contains that item. You can copy items (such as e-mail addresses) to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

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**.

Note: The Host Info panel is not displayed when conducting a web service scan.

In the following illustration, selecting **Cookies** displays a list of all sessions in which cookies were detected. If you select an item from the list, OpenText DAST displays the cookies associated with the selected session.

Host Info Panel Image



Options available

The Host Info options are described in the following table.

Option	Description
P3P Info	Displays Platform for Privacy Preferences Project (P3P) information.

Option	Description
	For more information, see "P3P info" below.
AJAX	Displays a list of all pages containing an AJAX engine, as well as the AJAX requests. For more information, see "AJAX" on the next page.
Certificates	Displays a list of all certificates associated with the site. For more information, see "Certificates" on page 83.
Comments	Displays a list of all URLs containing comments. For more information, see "Comments - host info" on page 83.
Cookies	Displays a list of all URLs containing cookies. For more information, see "Cookies" on page 83.
E-Mails	Displays a list of all URLs containing e-mail addresses in the response. For more information, see "E-mails - Host Info" on page 84.
Forms	Displays a list of all URLs containing forms. For more information, see "Forms - Host Info" on page 84.
Hiddens	Displays a list of all URLs containing input elements whose control type is "hidden." For more information, see "Hiddens - Host Info" on page 85.
Scripts	Displays a list of all URLs containing client-side scripts embedded in the server's response. For more information, see "Scripts - Host Info" on page 85.
Broken Links	Displays a list of all URLs containing hyperlinks to missing targets. For more information, see "Broken links" on page 86.
Offsite Links	Displays a list of all URLs containing hyperlinks to other sites. For more information, see "Offsite links" on page 86.
Parameters	Displays a list of all URLs containing embedded parameters. For more information, see "Parameters" on page 86.

P3P info

This option displays Platform for Privacy Preferences Project (P3P) information.

Note: According to the World Wide Web Consortium, work on P3P has been discontinued.

The World Wide Web Consortium's P3P enables websites 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 website 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 website whose policy states that a cookie is used for this purpose, the browser automatically rejects the cookie.

P3P user agents

Microsoft Internet Explorer 6 can display P3P privacy policies and compare the P3P policy with your own settings to decide whether or not to allow cookies from a particular site.

The Privacy Bird (originally developed by AT&T), which you can find at http://www.privacybird.com/, is a fully featured P3P user agent that automatically searches for privacy policies at every website the user visits. It then compares the policy with the user's stored privacy preferences and notifies the user of any discrepancies.

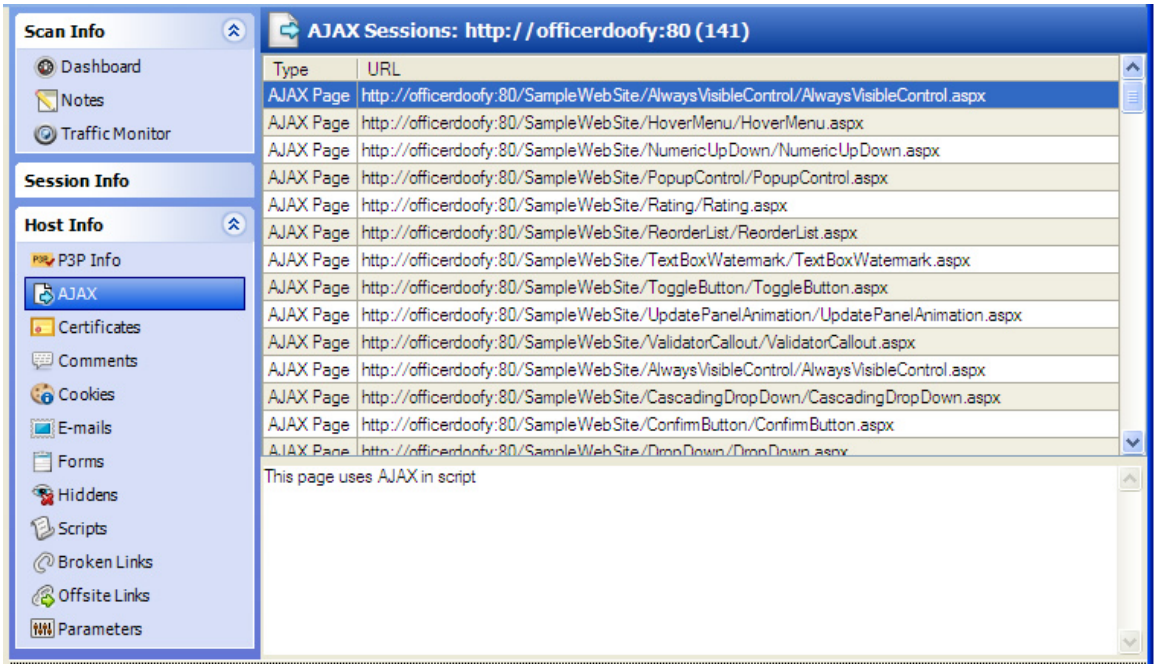
See also

"Host Info panel" on page 79

AJAX

AJAX is an acronym for Asynchronous JavaScript and XMLHttpRequest.

If you select this option, OpenText DAST displays all pages containing an AJAX engine, as well as the AJAX requests.



There are two types of AJAX line items in this view:

- AJAX Page (as illustrated above)
- Request

If you click an item in the list, OpenText DAST displays "This page uses AJAX in script" (for a Page type) or it lists the query and/or POST data parameters (for a Request type).

How AJAX works

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.

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, without stalling a user's interaction with the application.

Certificates

A certificate states that a specific website is secure and genuine. It ensures that no other website can assume the identity of the original secure site. A security 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.

Comments - host info

Developers sometimes leave critical information in comments that can be used to breach the security of a site. 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 compromise the security of your site.

To view discovered comments:

1. Select **Comments** from the **Host Info** panel to list all URLs that contain comments.
2. Click a **URL** to view the comments it contains.
3. Double-click an entry to locate in the navigation pane the session that contains the comment. Focus switches to the **Comments** choice in the **Session Info** panel.

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**.

You can copy comments to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

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.

To view discovered cookies:

1. Select **Cookies** from the **Host Info** panel to list all URLs in which cookies were found during a crawl or audit.
2. Click a URL to view the cookies it contains.
3. Double-click an entry to locate in the navigation pane the session that contains the cookie. Focus switches to the **HTTP Response** choice in the **Session Info** panel.

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**.

You can copy cookie code to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

E-mails - Host Info

OpenText DAST lists all email addresses discovered during a scan. To view the email addresses:

1. Select **E-mail** from the **Host Info** panel to list all URLs that contain email addresses.
2. Click a URL to view the email addresses it contains.
3. Double-click an entry to locate in the navigation pane the session that contains the email address. Focus switches to the **E-mail** choice in the **Session Info** panel.

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**.

You can copy email addresses to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

Forms - Host Info

OpenText DAST lists all HTML forms discovered during a scan.

1. Select **Forms** from the **Host Info** panel to list all URLs that contain forms.
2. Click a URL to view the source HTML of the form it contains.
3. Double-click an entry to locate in the navigation pane the session that contains the form. Focus switches to the Forms choice in the **Session Info** panel.

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**.

You can copy forms to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

Hiddens - Host Info

OpenText DAST 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.

1. Select **Hiddens** from the **Host Info** panel to list all URLs that contain hidden controls.
2. Click a URL to view the name and value attributes of the "hidden" controls contained in that URL.
3. Double-click an entry to locate in the navigation pane the session that contains the hidden control. Focus switches to the **Hiddens** choice in the **Session Info** panel.

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**.

You can copy the HTML text to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

Scripts - Host Info

OpenText DAST lists all scripts discovered during a scan. To view the discovered scripts:

1. Select **Scripts** from the **Host Info** panel to list all URLs that contain scripts.
2. Click a URL to view the script it contains.
3. Double-click an entry to locate in the navigation pane the session that contains the script.

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**.

You can copy a script to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

For more information on the OpenText DAST window, see "OpenText DAST user interface" on page 36.

See also

"Host Info panel" on page 79

"Navigation pane" on page 46

"Regular expressions" on page 310

Broken links

OpenText DAST finds and documents all non-working hyperlinks on the site. To locate broken links:

1. Select **Broken Links** from the **Host Info** panel to list all URLs that contain non-working hyperlinks.
2. Double-click an entry to locate in the navigation pane the session that contains a broken link. Focus switches to the **HTTP Response** choice in the **Session Info** panel.

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**.

You can copy the HTML text to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

Offsite links

OpenText DAST finds and documents all hyperlinks to other sites.

To examine hyperlinks to other sites:

1. Select **Offsite Links** from the **Host Info** panel to list all URLs that contain hyperlinks to other sites.
2. Double-click an entry to locate in the navigation pane the session that contains the offsite link. Focus switches to the **HTTP Response** choice in the **Session Info** panel.

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**.

You can copy the HTML text to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the session that contains the URL.

For more information on the OpenText DAST window, see "OpenText DAST user interface" on page 36.

Parameters

A parameter can be either of the following:

- A query string submitted as part of the URL in the HTTP request (or contained in another header).

- Data submitted using the Post method.

To list all URLs that contain parameters:

1. Select **Parameters** from the **Host Info** panel.
2. Click a URL to view the parameters it contains.
3. Double-click an entry to locate in the navigation pane the session that contains the parameter. For more information, see "Navigation pane" on page 46.

Use the **Search** feature at the top of the information pane to search the selected URL for the text you specify. To conduct a search using regular expressions, select the **Regex** button before clicking **Find**. For more information, see "Regular expressions" on page 310.

You can copy text to your clipboard by highlighting the text and selecting **Copy** from the shortcut menu.

If you double-click a URL, OpenText DAST highlights in the navigation pane the Session that contains the URL.

For more information, see "OpenText DAST user interface" on page 36.

See also

"Host Info panel" on page 79

Summary pane

When conducting or viewing a scan, use the horizontal summary pane at the bottom of the window to view a centralized display of vulnerable resources, quickly access vulnerability information, and view OpenText DAST logging information.

This pane has the following tabs:

- Findings (see "Findings tab" on the next page)
- Not Found (see Not Found tab)
- Scan Log (see "Scan Log tab " on page 92)
- Server Information (see "Server Information tab" on page 93)

See also

"OpenText DAST user interface" on page 36

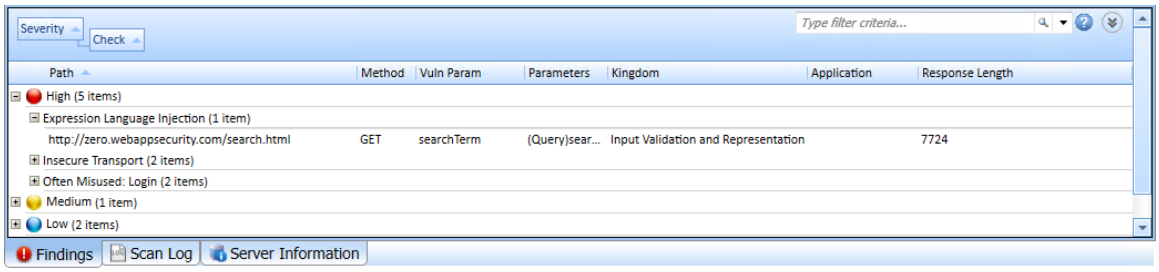
"Using filters and groups in the Summary pane" on page 244

"Retesting vulnerabilities" on page 226

"Vulnerability rollup" on page 253

Findings tab

The **Findings** tab lists information about each vulnerability discovered during an audit of your web application.



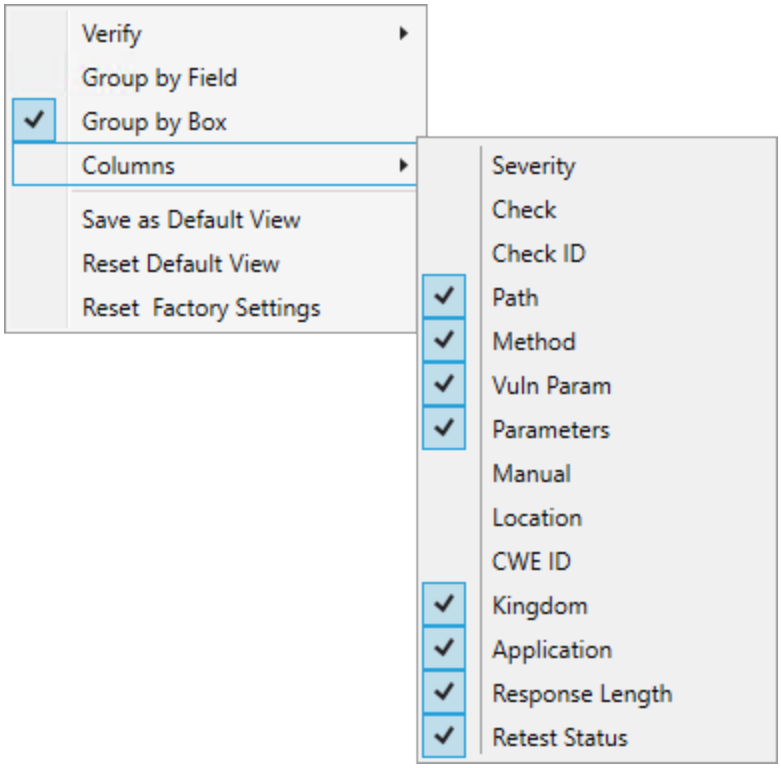
This tab also includes Informational issues discovered during the scan. These are not considered vulnerabilities, but identify interesting points in the site or certain applications or web servers.

Additionally, this tab includes Best Practices issues discovered during the scan. Likewise, these are not considered vulnerabilities, but relate to commonly accepted best practices for web development, and are indicators of overall site quality and site development security practices (or lack thereof).

Note: You can also group and filter results on the **Findings** tab. For more information, see "Using filters and groups in the Summary pane" on page 244.

Available columns

Several columns of data are available for viewing. 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**: An OpenText DAST probe for a specific vulnerability, such as cross-site scripting, unencrypted log-in form, etc.
- **Check ID**: The identification number of an OpenText DAST 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 OpenText DAST Agent. Column is available only when OpenText DAST Agent is enabled during 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 OpenText DAST Agent that are traceable to the same source. Column is available only when OpenText DAST Agent is enabled

during scan.

- **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 OpenText Software Security Research Group.
- **Application**: The application or framework in which the vulnerability is found, such as ASP.NET or Microsoft IIS server.
- **Pending Status**: The status (assigned automatically by OpenText DAST or manually) if this scan were to be published.
- **Published Status**: The status as it exists in Application Security, if previously published.
- **Reproducible**: Values may be Reproduced, Not Found/Fixed, or New. Column is available for Site Retests only (Retest Vulnerabilities).
- **Response Length**: The response size in bytes for the vulnerable session.
- **Retest Status**: The status of a verification scan that was performed on one or more issues. This column is available for a retest scan only. For more information, see "Retesting vulnerabilities" on page 226.

Vulnerability severity

The severity of vulnerabilities in the **Findings** tab is indicated by the following icons.

Critical	High	Medium	Low
			

Working with findings

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. For more information, see "Navigation pane" on page 46 and "Information pane" on page 57.

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.

If you right-click an item in the list, a shortcut menu enables 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** - 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: 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 244.

- **Change Severity** - Enables you to change the severity level.
- **Edit Vulnerability** - Displays the Edit Vulnerabilities dialog box, allowing you to modify various vulnerability characteristics. For more information, see "Editing vulnerabilities" on page 251.
- **Rollup Vulnerabilities** - Available if multiple vulnerabilities are selected; enables you to roll up the selected vulnerabilities into a single instance that is prefixed with the tag “[Rollup]” in OpenText DAST and reports. For more information, see "Vulnerability rollup" on page 253.

  Note: If you have selected a rolled up vulnerability, this menu option is **Undo Rollup Vulnerabilities**.

- **Retest** - Performs a retest of one or more selected vulnerabilities, all vulnerabilities, or vulnerabilities of a specific severity. For more information, see "Retesting vulnerabilities" on page 226.
- **Mark as** - Flags the vulnerability as either a false positive (and enables you to add a description) or as ignored. In both cases, the vulnerability is removed from the list. You can view a list of all false positive and ignored vulnerabilities by selecting **Suppressed Findings** in the Scan Info panel.

  Note: You can change a suppressed finding back to a vulnerability. See "Suppressed findings" on page 67 for details.

- **Send to** - Converts the vulnerability to a defect and adds it to the OpenText Application Lifecycle Management (ALM) 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 sessions" on page 258 for details.

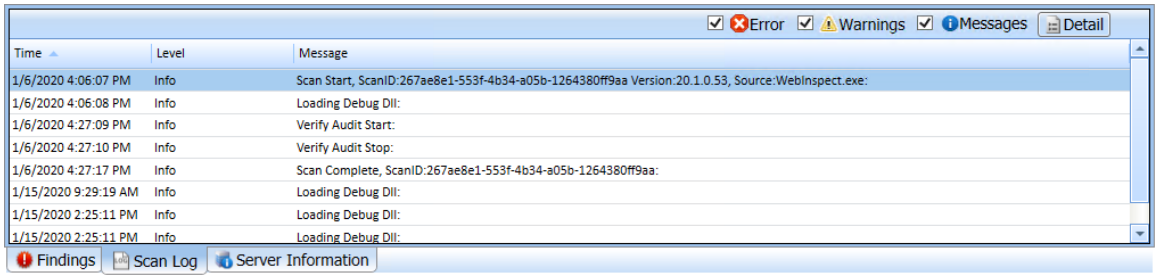
- **Crawl** - Recrawls the selected URL.
- **Tools** - Presents a submenu of available tools.
- **Attachments** - Enables you to create a note associated with the selected session, flag the session for follow-up, add a vulnerability note, or add a vulnerability screenshot.

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

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

Scan Log tab

Use the **Scan Log** tab to view information about your OpenText DAST scan activity. For instance, the time at which certain audit methodologies are applied against your web application are listed here. Additionally, alert-level messages that may provide insight into potential issues that could affect the scan appear in the Scan Log.



You can select the logging level (Debug, Info, Warn, Error, or Fatal) using the Logging option on the Application Settings window. For more information, see "Application settings: Logging" on page 450.

You can filter the type of messages displayed using the **Errors**, **Warnings**, and **Messages** buttons at the top of the pane.

Tip: Alert-level messages are included in the **Warnings** filter.

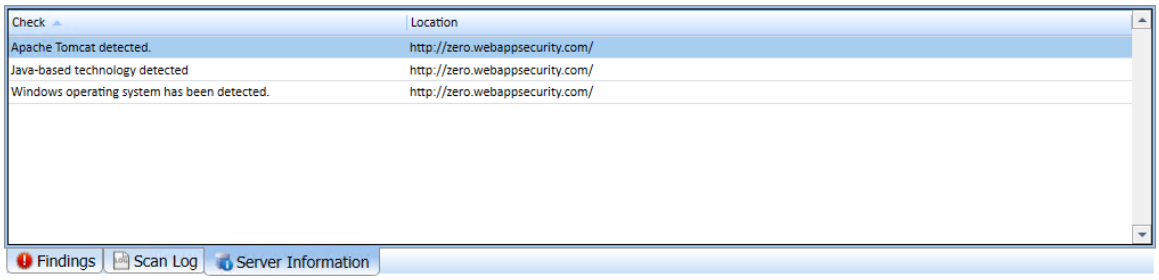
To view detailed information about a specific entry in the scan log, select an entry and then click **Detail**.

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

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

Server Information tab

The Server Information tab lists items of interest pertaining to the server. Only one occurrence of an item or event is listed per server.



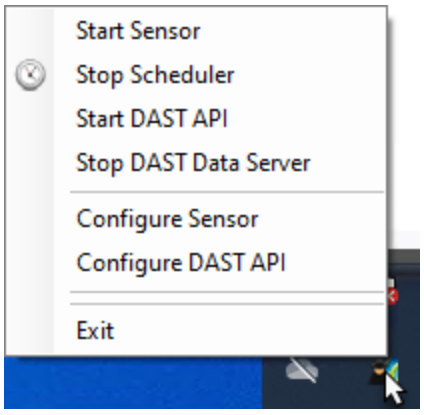
OpenText DAST Monitor

The OpenText DAST Monitor program, represented by an icon in the notification area of the taskbar, provides a context menu that enables you to:

- Start or stop the sensor service
- Start or stop the scheduler service
- Start or stop the DAST Data Server service

  Important! The DAST Data Server service is vital for OpenText DAST scans, as it is used for the Traffic Monitor database which is collected during a scan. It is also involved with other features, such as automatic logout detection, the Web Proxy tool, Traffic Viewer tool, and the FAST proxy. Do not stop the DAST Data Server unless instructed to do so by Customer Support.

- Configure and start or stop the OpenText DAST API



Pop-up messages also appear whenever certain events occur.

Chapter 4: Working with scans

This chapter describes the various types of scans that OpenText DAST can perform, as well as instructions on how to run those scans. It includes procedures for scheduling scans, and importing, exporting, and managing scans that have completed.

Guided Scan overview

Guided Scan directs you through the best steps for configuring a scan tailored to your application.

The Guided Scan progress display in the left pane enables you to easily see your progress as you specify settings for your scan. The right pane displays the scan options on each wizard page.

The Guided Scan Wizard enables you to:

- Verify connectivity to your application
- Test the entire application or only workflows
- Record your login procedure
- Review suggested configuration changes

Guided Scans are template based; you can select to use either a Predefined Template or a Mobile Template.

Predefined templates

There are three predefined templates options to choose from:

- **Standard Scan**: use this option to when you are interested in coverage. Larger sites could take days when using this template.
- **Quick Scan**: use this option when focusing on breadth and performance rather than digging deep. Especially good for very large sites.
- **Thorough Scan**: use to perform an exhaustive crawl on your site. It is recommended that you split your site into parts and only scan smaller chunks of your site with these settings. Not recommended for large sites.

Mobile templates

There are two mobile template options to choose from:

- **Mobile Scan**: use this option to scan a mobile site from the machine where your instance of OpenText DAST is installed. OpenText DAST will fetch the mobile version

of the site rather than the full site when this option is chosen.

- **Native Scan**: use this option to manually crawl a native mobile application and capture the web traffic as a workflow macro. Generate the traffic on an Android, Windows, or iOS device or software emulator (Android and iOS only) running a mobile application.

After selecting a Guided Scan template, the stages and steps are displayed in the left pane, allowing you to easily navigate among them and specify the settings for your scan.

See also

"Using the Predefined template" on the next page

"Using the Mobile Scan template" on page 110

"Using the Native Scan template" on page 123

Running a Guided Scan

The Guided Scan progress display in the left pane enables you to easily see your progress as you specify settings for your scan. The right pane displays the scan options on each wizard page.

The first page of the Guided Scan presents you with the option to select the type of scan to run. There are three main types to choose from.

Predefined template (Standard, Quick, or Thorough)

There are three Predefined templates options to choose from:

- **Standard Scan:** Default scan settings are designed to focus more on coverage than performance. Larger sites could take days to crawl with these settings.
- **Quick Scan:** A scan that focuses on breadth and performance rather than digging deep. Especially good for very large sites.
- **Thorough Scan:** Thorough scan settings are designed to perform an exhaustive crawl of your site. It is recommended that your split your site up into parts and only scan smaller chunks of your site with these settings. Not recommended for large sites.

For more information, see "Using the Predefined template" on the next page.

Mobile scan template

This template emulates a mobile device while scanning a web application.

For more information, see "Using the Mobile Scan template" on page 110.

Native scan template

This template manually crawls a native mobile application and captures web traffic as a workflow macro.

For more information, see "Using the Native Scan template" on page 123.

See also

"Guided Scan overview " on page 95

"OpenText DAST policies" on page 458

Using the Predefined template

The Guided Scan wizard will step you through the necessary stages and steps required to scan your website. If you need to return to a previous step or stage, click the back navigation button, or click the step in the Guided Scan tree to be taken directly there.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Launching a Guided Scan

To launch a Guided Scan:

- For OpenText DAST users, click the Start a Guided Scan option in the left pane, or select **File > New > Guided Scan** from the menu bar.

The Guided Scan wizard launches and presents a list of Guided Scan templates. There are three Predefined templates options to choose from:

- **Standard Scan**: use this option to when you are interested in coverage. Larger sites could take days when using this template.
- **Quick Scan**: use this option when focusing on breadth and performance rather than digging deep. Especially good for very large sites.
- **Thorough Scan**: use to perform an exhaustive crawl on your site. It is recommended that you split your site into parts and only scan smaller chunks of your site with these settings. Not recommended for large sites.

Choose one of the **Predefined Templates**.

Understanding the Rendering Engine

The Rendering Engine you select determines which Web Macro Recorder is opened when recording a new macro or editing an existing macro while configuring a Guided Scan. The Rendering Engine options are:

- **Session-based** – Selecting this option designates the Session-based Web Macro Recorder, which uses Internet Explorer browser technology.
- **Event-based (preferred)** – Selecting this option designates the Event-based Web Macro Recorder, which uses TruClient and Firefox technology.

Verifying your website

To verify your website:

1. In the Start URL box, type or select the complete URL or IP address of the site to scan.

   If you enter a URL, it must be precise. For example, if you enter MYCOMPANY.COM, OpenText DAST 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 results 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 do not pursue links that use fully qualified URLs (as opposed to relative paths).

   Note: OpenText DAST supports Internet Protocol version 6 (IPv6) addresses in website and web service scans. When you specify the Start URL, you must enclose the IPv6 address in brackets. For example:

   - http://[::1]
     OpenText DAST scans "localhost."
   - http://[fe80::20c:29ff:fe32:bae1]/subfolder/
     OpenText DAST scans the host at the specified address starting in the "subfolder" directory.
   - http://[fe80::20c:29ff:fe32:bae1]:8080/subfolder/
     OpenText DAST scans a server running on port 8080 starting in "subfolder."

   OpenText DAST supports both Internet Protocol version 4 (IPV4) and Internet Protocol version 6 (IPV6). IPV6 addresses must be enclosed in brackets.

2. (Optional) To limit the scope of the scan to an area, select the **Restrict to Folder** check box, and then select one of the following options from the list:

   Directoryonly (self). OpenText DAST 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/, OpenText DAST will assess only the "two" directory.

   Directory and subdirectories. OpenText DAST 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. OpenText DAST will begin crawling and/or auditing at the URL you specify, but will not access any directory that is lower in the directory tree.

   For information about limitations to the Restrict to folder scan option, see "Restrict to folder limitations" on page 196.

3. Click **Verify**.

   If the website is set up to be authenticated with a client certificate using a common access card (CAC) or a certificate that is password protected, then Guided Scan will prompt you with the following message:

   The site *<URL>* is requesting a client certificate. Would you like to configure one now?

   To configure a client certificate that uses a CAC or a certificate that is password protected:

   a. Click **Yes**.

      The Select a Client Certificate window appears.

   b. Under Certificate Store, select **Current User**.

      A list of available certificates appears in the Certificate area.

   c. Locate and select a certificate that is prefixed with “(Protected)”.

      Information about the selected certificate and a Password/PIN field appear in the Certificate Information area.

   d. If a password or PIN is required, type it in the **Password/PIN** field.

      Note: If a password or PIN is required and you do not enter it at this point, you must enter the password or PIN in the Windows Security window each time it prompts you during the scan.

      Important! By default, OpenText DAST uses OpenSSL. If you are using a specific SSL/TLS protocol rather than OpenSSL, the Profiler portion of scan configuration may not work with certificates that are protected with a password.

   e. Click **Test**.

4. If you must access the target site through a proxy server, click **Proxy** in the lower left of the main screen to display the Proxy Settings area, and then select an option from the **Proxy Settings** list:
   - **Direct Connection (proxy disabled)**
   - **Auto detect proxy settings**: Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's Web proxy settings.
   - **Use System proxy settings**: Import your proxy server information from the local machine.
   - **Use Firefox proxy settings**: Import your proxy server information from Firefox.
   - **Configure proxy settings using a 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.
   - **Explicitly configure proxy settings**: Specify proxy server settings as indicated. If you select this option, enter the proxy information in the fields provided.

   Important! Socks4 proxy servers do not support authentication. When using a Socks proxy server that requires authentication, you must use a Socks5 proxy.

   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 Windows setting "Use a proxy server for your LAN" is not selected, then a proxy server is not used.

   When a screenshot of the website or directory structure appears, you have successfully verified your connection to the Start URL.
5. Click **Next**.

   The Choose Scan Type window appears.

Choosing a scan type

1. Type in a name for your scan in the **Scan Name** box.
2. Select one of the following scan types:
   - **Standard**: OpenText DAST performs an automated analysis, starting from the target URL. This is the normal way to start a scan.
   - **Workflows:** If you select this option, an additional Workflows stage is added to the Guided scan.
3. In the Scan Method area, select one of the following scan methods:
   - **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**. OpenText DAST 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 "Scan settings: Method" on page 358.
- **Audit Only**. OpenText DAST applies the methodologies of the selected policy to determine vulnerability risks, but does not crawl the website. No links on the site are followed or assessed.

4. In the Policy area, select a policy from the Policy list. For information about managing policies, see the Policy Manager chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
5. In the Crawl Coverage area, select the level of coverage you want using the **Crawl Coverage** slider. For more information on crawl coverage levels, see "Configuring crawl coverage and thoroughness" on page 179.
6. In the **Single-Page Applications** area, select an option for crawling and auditing single-page applications (SPAs). When enabled, the DOM script engine finds JavaScript includes, frame and iframe includes, CSS file includes, and AJAX calls during the crawl, and then audits all traffic generated by those events. Options for Single-Page Applications are:
   - **Automatic** - If OpenText DAST detects a SPA framework, it automatically switches to SPA-support mode.
   - **Enabled** - Indicates that SPA frameworks are used in the target application.

     Caution! SPA support should be enabled for single-page applications only. Enabling SPA support to scan a non-SPA website will result in a slow scan.
   - **Disabled** - Indicates that SPA frameworks are not used in the target application.

   For more information, see "About single-page application scans" on page 201.
7. Click the **Next** button.
   The Login stage appears with Network Authentication highlighted in the left pane.

Configuring network authentication

If your network requires user authentication, you can configure it here. If your network does not require user authentication, click the Next navigation button or the next appropriate step in the Guided Scan tree to continue on.

To configure network authentication:

1. Click the **Network Authentication** checkbox.
2. Select a **Method** from the drop-down list of authentication methods. The authentication methods are:
   - **ADFS CBT**
   - **Automatic**
   - **Basic**
   - **Digest**
   - **Kerberos**
   - **Negotiate**
   - **NT LAN Manager (NTLM)**
   - **OAuth 2.0 Bearer**
3. Do one of the following:
   - For all authentication methods *except* OAuth 2.0 Bearer, type a user ID in the **User Name** box and the user's password in the **Password** box.
   - For the OAuth 2.0 Bearer method, click **Configure** and continue with "Configuring OAuth 2.0 bearer credentials" on page 400.

Using a client certificate

To use a client certificate for network authentication:

1. To use a client certificate for network authentication, select **Client Certificate**.
2. In the Certificate Store area, select one of the following, and then select either the **My** or **Root** radio button:
   - **Local Machine**. OpenText DAST uses a certificate on the local machine based on your selection in the Certificate Store area.
   - **Current User**. OpenText DAST uses a certificate for the current user based on your selection in the Certificate Store area.
3. To view certificate details in the Certificate Information area, select a certificate.
4. Click the **Next** button.

   The Application Authentication page appears.

Configuring application authentication

If your site requires authentication, you can use this step to create, select, or edit a login macro to automate the login process and increase the coverage of your site. A login macro is a recording of the activity that is required to access and log in to your

application, typically by entering a user name and password and clicking a button such as Log In or Log On.

If **Enable macro validation** is selected in Scan Settings: Authentication for scans that use a login macro, OpenText DAST tests the login macro at the start of the scan to ensure that the log in is successful. If the macro is invalid and fails to log in to the application, the scan stops and an error message is written in the scan log file. For more information and troubleshooting tips, see "Testing login macros" on page 494.

Note: Macro testing is not supported for macros containing two-factor authentication.

Important! If you use a macro that includes Two-factor Authentication, then you must configure the Two-factor Authentication Application settings before starting the scan. For more information, see "Application settings: Two-Factor Authentication" on page 442.

The following options are available for login macros:

- "Using a login macro without privilege escalation " below
- "Using login macros for privilege escalation" on the next page

Masked values supported

If the macro uses parameters for which values are masked in the Web Macro Recorder, then these values are also masked when configuring a Guided Scan in OpenText DAST.

Using a login macro without privilege escalation

To use a login macro:

1. Select the **Use a login macro for this site** check box.
2. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
3. Click the **Next** button.

   If you selected a Standard scan, the Optimization Tasks page appears. If you selected a Workflows scan, the Manage Workflows page appears.

Using login macros for privilege escalation

If you selected the Privilege Escalation policy or another policy that includes enabled Privilege Escalation checks, at least one login macro for a high-privilege user account is required. For more information, see "About privilege escalation scans" on page 198.

To use login macros:

1. Select the **High-Privilege User Account Login Macro** check box. This login macro is for the higher-privilege user account, such as a Site Administrator or Moderator account.
2. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

   After recording or selecting the first macro and clicking the next arrow, a "Configure Low Privilege Login Macro" prompt appears.
3. Do one of the following:
   - To perform the scan in authenticated mode, click **Yes**. For more information, see "About privilege escalation scans" on page 198.

     Guided Scan returns to the Select Login Macro window for you to create or select a low-privilege login macro. Continue to Step 4.
   - To perform the scan in unauthenticated mode, click **No**. For more information, see "About privilege escalation scans" on page 198.

     The Application Authentication Step is complete. If you selected a Standard scan, the Optimization Tasks page appears. If you selected a Workflows scan, the Manage Workflows page appears.
4. Select the **Low-Privilege User Account Login Macro** check box. This login macro is for the lower-privilege user account, such as a viewer or consumer of the site content.
5. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

6. After recording or selecting the second macro, click the **Next** button.

   If you selected a Standard scan, the Optimization Tasks page appears. If you selected a Workflows scan, the Manage Workflows page appears.

Automatically creating a login macro

You can enter a username and password and have OpenText DAST create a login macro automatically.

Note: You cannot automatically create login macros for privilege-escalation and multi-user login scans or for any scan using the Session-based rendering engine.

To automatically create a login macro:

1. Select **Auto-gen Login Macro**.
2. Type a username in the **Username** field.
3. Type a password in the **Password** field.

Optionally, click **Test** to locate the login form, generate the macro, and run macro validation tests before advancing to the next stage in the Guided Scan wizard. If you need to cancel the validation test prior to completion, click **Cancel**.

If the macro is invalid and fails to log in to the application, an error message appears. For more information and troubleshooting tips, see "Testing login macros" on page 494.

Configuring workflows

The Workflows stage only appears if you selected Workflows as the Scan Type in the Site stage. If you chose Standard, the Workflows stage will not appear. You can create a Workflow macro to ensure OpenText DAST audits the pages you specify in the macro. OpenText DAST audits only those URLs included in the macro 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.

Important! If you use a login macro in conjunction with a workflow macro or startup macro or both, all macros must be of the same type: all `.webmacro` files or all Burp Proxy captures or all `.har` files. You cannot use different types of macros in the same scan.

To complete the Workflows settings, click any of the following in the Workflows table:

- **Record**. Opens the Web Macro Recorder, allowing you to create a macro.
- **Edit**. Opens the Web Macro Recorder and loads the selected macro.
- **Delete**. Removes the selected macro (but does not delete it from your disk).

- **Import**. Opens a standard file-selection window, allowing you to select a previously recorded `.webmacro` file, Burp Proxy captures, or `.har` file.

  Note: If you have installed OpenText Unified Functional Testing (UFT One) on your computer, then OpenText DAST detects this automatically and displays an option to import a UFT `.usr` file.

  For more information, see "Importing Functional Testing files in a Guided Scan" on page 135.

- **Export**. Opens a standard file-selection window, allowing you to save a recorded macro.

After you specify and play a workflow macro, it appears in the Workflows table and its Allowed Hosts are added to the **Guided Scan > Workflows > Workflows > Manager Workflow** page. You can enable or disable access to particular hosts. For more information, see "Scan settings: Allowed Hosts" on page 375.

Adding Burp Proxy results

If you have run Burp Proxy security tests, the traffic collected during those tests can be imported into a Workflow macro, reducing the time it would otherwise take to rescan the same areas.

To add Burp Proxy results to a workflow macro:

1. If you are not on the Workflows screen, click on the **Manage Workflows** step in the Guided Scan tree.
2. Click the **Import** button.

   The Import Macro file selector appears.
3. Change the file type box filter from Web Macro (*`.webmacro`) to Burp Proxy (*.*).
4. Navigate to your Burp Proxy files and select the desired file.
5. Click **Open**.

Using the Profiler

The OpenText DAST Profiler conducts a preliminary examination of the target website 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 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 Profiler's suggestion to configure the required information before continuing.

Similarly, your settings may specify that OpenText DAST should not conduct "file-not-found" detection. This process is useful for websites 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 OpenText DAST setting to accommodate this feature.

To launch the Profiler:

1. Click **Profile**.

   The Profiler runs. For more information, see "Server Profiler" on page 238.

   Results appear in the Optimize scan for box in the Settings section.
2. Accept or reject the suggestions that appear in the Optimize scan for drop-down box. To reject the suggestion, select None or an alternate from the drop-down menu.
3. If necessary, provide any requested information.
4. Click the **Next** button.

Several options may be presented even if you do not run the Profiler, as described in the following sections.

Autofill web forms

Select **Auto-fill Web forms during crawl** if you want OpenText DAST to submit values for input controls on forms it encounters while scanning the target site. OpenText DAST will extract the values from a prepackaged default file or from a file that you create using the Web Form Editor. See the Web Form Editor chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*. You may:

1. Click the ellipsis button **(...)** to locate and load a file.
2. Click **Edit** to edit the selected file (or the default values) using the Web Form Editor.
3. 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. For more information, see "Scan settings: Allowed Hosts" on page 375.

To add allowed domains:

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

Reuse identified Suppressed Findings

You can import vulnerabilities that were changed to false positive or ignored in previous scans. If those false positive or ignored items match vulnerabilities detected in the current scan, the vulnerabilities will be changed to false positive or ignored. You can import suppressed findings from existing scans or suppressed findings files. For more information, see "Suppressed findings" on page 67.

To reuse identified suppressed findings:

1. Select **Import Suppressed Findings**.
2. Continue according to the following table.

To use...	Then...
Existing scans	a. Click **Click here to import suppressed findings from scans**. The Select a Scan to Import Suppressed Findings dialog opens. b. Select one or more scans containing suppressed findings from the same site you are now scanning. c. Click **OK**.
Suppressed findings files	a. Click **Click here to import suppressed findings from a file**. A standard Windows file selection dialog box opens. b. Select the file to import, and then click **Open**. c. Optionally, repeat Steps a and b to select additional files.

Apply sample macro

OpenText'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 OpenText DAST and the responses returned by the target server.

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

Message

If the Profiler does not recommend changes, the Guided Scan wizard displays the message "No settings changes are recommended. Your current scan settings are optimal for this site."

Click **Next**.

The Final Review page appears with **Configure Detailed Options** highlighted in the left pane.

Configuring additional options

To configure detailed options, specify any of the following settings.

Reuse identified False Positives

Select the **False Positives** box to reuse false positives that OpenText DAST has already identified.

Traffic Analysis

1. To use the Web Proxy tool, select **Launch and Direct Traffic through Web Proxy** to use the Web Proxy tool to examine the HTTP requests issued by OpenText DAST and the responses returned by the target server.

   Web Proxy is a stand-alone, self-contained proxy server that you can configure and run on your desktop. Web Proxy enables you to monitor traffic from a scanner, a web browser, or any other tool that submits HTTP requests and receives responses from a server. Web Proxy is a tool for a debugging and penetration scan; you can view every request and server response while browsing a site.
2. Select the **Traffic Monitor** box to display and review each HTTP request sent by OpenText DAST and the associated HTTP response received from the server.

   While scanning a website, OpenText DAST displays only those sessions that reveal the hierarchical structure of the website, plus those sessions in which a vulnerability was discovered. However, if you select **Enable Traffic Monitor**, OpenText DAST enables you to display and review each HTTP request sent by OpenText DAST and the associated HTTP response received from the server.
3. Click **Next**.

   The Validate Settings and Start Scan page appears with Configure Detailed Options highlighted in the left pane.

Validating settings and starting the scan

Options on this page allow you to save the current scan settings.

1. To save your scan settings as an XML file, select **Click here to save settings**. Use the standard Save as window to name and save the file.
2. In the **Scan Now** area, review your scan settings, and then click **Start Scan** to begin the scan.

See also

"Guided Scan overview " on page 95

Using the Mobile Scan template

Using the Mobile Scan template to create a mobile website scan enables you to scan the mobile version of a website using the desktop version of your browser from within OpenText DAST.

A Mobile Scan is nearly identical to a website scan and mirrors the settings options you will find when using one of the Predefined templates to do a Standard, Thorough, or Quick scan. The only difference is that you need to select a user agent header to allow your browser to emulate a mobile browser.

OpenText DAST comes with four mobile user agent options to choose from, but you can create a custom option and create a user agent for another version of Android, Windows Phone, or other mobile device. For information creating a user agent header, see "Creating a custom user agent header" on the next page.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Launching a Mobile Scan

To launch a Mobile Scan:

1. click **Start a Guided Scan** on the OpenText DAST Start page.
2. Select **Mobile Scan** from the Mobile Templates section.
3. Click the **Mobile Client** icon in the tool bar.
4. Select the **Rendering Engine** you want to use. The rendering engine you select determines which Web Macro Recorder is opened when recording a new macro or editing an existing macro while configuring a Guided Scan. The rendering engine options are:
   - **Session-based** – Selecting this option designates the Session-based Web Macro Recorder, which uses Internet Explorer browser technology.
   - **Event-based (preferred)** – Selecting this option designates the Event-based Web Macro Recorder, which uses TruClient and Firefox technology.
5. Select the User Agent that represents the agent string you want your rendering engine to present to the site. If you created your own user string, it will appear as Custom. If the user agent is not listed, you can create a custom user agent. See "Creating a custom user agent header" on the next page.

The Guided Scan wizard displays the first step in the Native Mobile Stage: Verify website.

Creating a custom user agent header

OpenText DAST includes user agents for Android, Windows, and iOS devices. If you are using one of these options, you do not need to create a custom user agent header. If you want your web browser to identify itself as a different mobile device or a specific OS version, create a custom user agent header.

To create a custom user agent:

1. Click the **Advanced** icon in the Guided Scan tool bar.
2. The Scan Settings window appears.
3. In the Scan Settings column, select **Cookies/Headers**.
4. In the Append Custom Headers section of the settings area, double-click the User-Agent string.

   The Specify Custom Header box appears.
5. Type in User-Agent: followed by the user agent header string for the desired device.
6. Click **OK**.

   The new custom user agent will now be available to select as your Mobile Client.

Verifying your website

To verify your website:

1. In the **Start URL** box, type or select the complete URL or IP address of the site to scan.

   If you enter a URL, it must be precise. For example, if you enter MYCOMPANY.COM, OpenText DAST 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 results 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 do not pursue links that use fully qualified URLs (as opposed to relative paths).

   OpenText DAST supports both Internet Protocol version 4 (IPV4) and Internet Protocol version 6 (IPV6). IPV6 addresses must be enclosed in brackets.

   Note: OpenText DAST supports Internet Protocol version 6 (IPv6) addresses in website and web service scans. When you specify the Start URL, you must enclose the IPv6 address in brackets. For example:

   - http://[::1]

     OpenText DAST scans "localhost."

- http://[fe80::20c:29ff:fe32:bae1]/subfolder/
  OpenText DAST scans the host at the specified address starting in the "subfolder" directory.
- http://[fe80::20c:29ff:fe32:bae1]:8080/subfolder/
  OpenText DAST scans a server running on port 8080 starting in "subfolder."

2. (Optional) To limit the scope of the scan to an area, select the **Restrict to Folder** check box, and then select one of the following options from the list:
   - Directory only (self). OpenText DAST 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/, OpenText DAST will assess only the "two" directory.
   - Directory and subdirectories. OpenText DAST 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. OpenText DAST will begin crawling and/or auditing at the URL you specify, but will not access any directory that is lower in the directory tree.

   For information about limitations to the Restrict to folder scan option, see "Restrict to folder limitations" on page 196.
3. Click **Verify**.
   If the website is set up to be authenticated with a client certificate using a common access card (CAC) or a certificate that is password protected, then Guided Scan will prompt you with the following message:
   The site *<URL>* is requesting a client certificate. Would you like to configure one now?
   To configure a client certificate that uses a CAC or a certificate that is password protected:
   a. Click **Yes**.
      The Select a Client Certificate window appears.
   b. Under Certificate Store, select **Current User**.
      A list of available certificates appears in the Certificate area.
   c. Locate and select a certificate that is prefixed with “(Protected)”.
      Information about the selected certificate and a Password/PIN field appear in the Certificate Information area.
   d. If a password or PIN is required, type it in the **Password/PIN** field.

      Note: If a password or PIN is required and you do not enter it at this point, you must enter the password or PIN in the Windows Security window each time it prompts you during the scan.

Important! By default, OpenText DAST uses OpenSSL. If you are using a specific SSL/TLS protocol rather than OpenSSL, the Profiler portion of scan configuration may not work with certificates that are protected with a password.

e. Click **Test**.

4. If you must access the target site through a proxy server, click **Proxy** in the lower left of the main screen to display the Proxy Settings area, and then select an option from the Proxy Settings list:
   - **Direct Connection (proxy disabled)**
   - **Auto detect proxy settings**: Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's web proxy settings.
   - **Use System proxy settings**: Import your proxy server information from the local machine.
   - **Use Firefox proxy settings**: Import your proxy server information from Firefox.
   - **Configure proxy settings using a 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.
   - **Explicitly configure proxy settings**: Specify proxy server settings as indicated. If you select this option, enter the proxy information in the fields provided.

   Important! Socks4 proxy servers do not support authentication. When using a Socks proxy server that requires authentication, you must use a Socks5 proxy.

   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 Windows setting "Use a proxy server for your LAN" is not selected, then a proxy server is not used.

   When a screenshot of the website or directory structure appears, you have successfully verified your connection to the Start URL.
5. Click **Next.**

   The Choose Scan Type window appears.

Choosing a scan type

1. Type in a name for your scan in the **Scan Name** box.
2. Select one of the following scan types:
   - **Standard**: OpenText DAST performs an automated analysis, starting from the target URL. This is the normal way to start a scan.

- **Workflows**: If you select this option, an additional Workflows stage is added to the Guided scan.

3. In the Scan Method area, select one of the following scan methods:
   - **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**: OpenText DAST 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 359.
   - **Audit Only**: OpenText DAST applies the methodologies of the selected policy to determine vulnerability risks, but does not crawl the website. No links on the site are followed or assessed.
4. In the Policy area, select a policy from the Policy list. For information about managing policies, see the Policy Manager chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
5. In the Crawl Coverage area, select the level of coverage you want using the **Crawl Coverage** slider. For more information on crawl coverage levels, see "Configuring crawl coverage and thoroughness" on page 179.
6. In the **Single-Page Applications** area, select an option for crawling and auditing single-page applications (SPAs). When enabled, the DOM script engine finds JavaScript includes, frame and iframe includes, CSS file includes, and AJAX calls during the crawl, and then audits all traffic generated by those events. Options for Single-Page Applications are:
   - **Automatic** - If OpenText DAST detects a SPA framework, it automatically switches to SPA-support mode.
   - **Enabled** - Indicates that SPA frameworks are used in the target application.

     Caution! SPA support should be enabled for single-page applications only. Enabling SPA support to scan a non-SPA website will result in a slow scan.
   - **Disabled** - Indicates that SPA frameworks are not used in the target application.

   For more information, see "About single-page application scans" on page 201.
7. Click the **Next** button.

   The Login stage appears with Network Authentication highlighted in the left pane.

Configuring network authentication

If your network requires user authentication, you can configure it here. If your network does not require user authentication, click the Next navigation button or the next appropriate step in the Guided Scan tree to continue on.

To configure network authentication:

1. Click the **Network Authentication** checkbox.
2. Select a **Method** from the drop-down list of authentication methods. The authentication methods are:
   - **ADFS CBT**
   - **Automatic**
   - **Basic**
   - **Digest**
   - **Kerberos**
   - **Negotiate**
   - **NT LAN Manager (NTLM)**
   - **OAuth 2.0 Bearer**
3. Do one of the following:
   - For all authentication methods except OAuth 2.0 Bearer, type a user ID in the **User Name** box and the user's password in the **Password** box.
   - For the OAuth 2.0 Bearer method, click **Configure** and continue with "Configuring OAuth 2.0 bearer credentials" on page 400.

Using client certificates

To use client certificates for network authentication:

1. To use a client certificate for network authentication, select **Client Certificate**.

   Note: You can add a client certificate to a Windows phone, but the only way to subsequently remove it is to restore the phone to its default settings.

2. In the Certificate Store area, select one of the following, and then select either the **My** or **Root** radio button:
   - **Local Machine**. OpenText DAST uses a certificate on the local machine based on your selection in the Certificate Store area.
   - **Current User**. OpenText DAST uses a certificate for the current user based on your selection in the Certificate Store area.
3. To view certificate details in the Certificate Information area, select a certificate.
4. Click the **Next** button.

   The Application Authentication page appears.

Configuring application authentication

If your site requires authentication, you can use this step to create, select, or edit a login macro to automate the login process and increase the coverage of your site. A login macro is a recording of the activity that is required to access and log in to your application, typically by entering a user name and password and clicking a button such as Log In or Log On.

If **Enable macro validation** is selected in Scan Settings: Authentication for scans that use a login macro, OpenText DAST tests the login macro at the start of the scan to ensure that the log in is successful. If the macro is invalid and fails to log in to the application, the scan stops and an error message is written in the scan log file. For more information and troubleshooting tips, see "Testing login macros" on page 494.

Note: Macro testing is not supported for macros containing two-factor authentication.

Important! If you use a macro that includes Two-factor Authentication, then you must configure the Two-factor Authentication Application settings before starting the scan. For more information, see "Application settings: Two-Factor Authentication" on page 442.

The following options are available for login macros:

- "Using a login macro without privilege escalation " below
- "Using login macros for privilege escalation" on the next page

Masked values supported

If the macro uses parameters for which values are masked in the Web Macro Recorder, then these values are also masked when configuring a Guided Scan in OpenText DAST.

Using a login macro without privilege escalation

To use a login macro:

1. Select the **Use a login macro for this site** check box.
2. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

3. Click the **Next** button.

   If you selected a Standard scan, the Optimization Tasks page appears. If you selected a Workflows scan, the Manage Workflows page appears.

Using login macros for privilege escalation

If you selected the Privilege Escalation policy or another policy that includes enabled Privilege Escalation checks, at least one login macro for a high-privilege user account is required. For more information, see "About privilege escalation scans" on page 198.

To use login macros:

1. Select the **High-Privilege User Account Login Macro** check box. This login macro is for the higher-privilege user account, such as a Site Administrator or Moderator account.
2. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

   After recording or selecting the first macro and clicking the next arrow, a "Configure Low Privilege Login Macro" prompt appears.
3. Do one of the following:
   - To perform the scan in authenticated mode, click **Yes**. For more information, see "About privilege escalation scans" on page 198.

     Guided Scan returns to the Select Login Macro window for you to create or select a low-privilege login macro. Continue to Step 4.
   - To perform the scan in unauthenticated mode, click **No**. For more information, see "About privilege escalation scans" on page 198.

     The Application Authentication Step is complete. If you selected a Standard scan, the Optimization Tasks page appears. If you selected a Workflows scan, the Manage Workflows page appears.
4. Select the **Low-Privilege User Account Login Macro** check box. This login macro is for the lower-privilege user account, such as a viewer or consumer of the site content.
5. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.

- To edit an existing login macro shown in the Login Macro field, click **Edit**.
- To record a new macro, click **Create**.

For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

6. After recording or selecting the second macro, click the **Next** button.

   If you selected a Standard scan, the Optimization Tasks page appears. If you selected a Workflows scan, the Manage Workflows page appears.

Automatically creating a login macro

You can enter a username and password and have OpenText DAST create a login macro automatically.

Note: You cannot automatically create login macros for privilege-escalation and multi-user login scans or for any scan using the Session-based rendering engine.

To automatically create a login macro:

1. Select **Auto-gen Login Macro**.
2. Type a username in the **Username** field.
3. Type a password in the **Password** field.

Optionally, click **Test** to locate the login form, generate the macro, and run macro validation tests before advancing to the next stage in the Guided Scan wizard. If you need to cancel the validation test prior to completion, click **Cancel**.

If the macro is invalid and fails to log in to the application, an error message appears. For more information and troubleshooting tips, see "Testing login macros" on page 494.

About the Workflows stage

The Workflows stage only appears if you selected Workflows as the Scan Type in the Site stage. If you chose Standard, the Workflows stage will not appear.

You can create a Workflow macro to ensure OpenText DAST audits the pages you specify in the macro. OpenText DAST audits only those URLs included in the macro and does not follow any hyperlinks encountered during the audit.

You can create multiple Workflow macros; one for each use case on your site. A logout signature is not required. This type of macro is used most often to focus on a particular subsection of the application. If you select multiple macros, they will all be included in the same scan. In addition to allowing you to select multiple macros, you can also import Burp proxy captures and `.har` files, and add them to your scan.

Important! If you use a login macro in conjunction with a workflow macro or startup macro or both, all macros must be of the same type: all `.webmacro` files or all Burp Proxy captures or all `.har` files. You cannot use different types of macros in the same scan.

To complete the Workflows settings, click any of the following in the Workflows table:

- **Record**. Opens the Web Macro Recorder, allowing you to create a macro.
- **Edit**. Opens the Web Macro Recorder and loads the selected macro.
- **Delete**. Removes the selected macro (but does not delete it from your disk).
- **Import**. Opens a standard file-selection window, allowing you to select a previously recorded `.webmacro` file, Burp Proxy captures, or `.har` file.

  Note: If you have installed OpenText Unified Functional Testing (UFT One) on your computer, then OpenText DAST detects this automatically and displays an option to import a UFT `.usr` file.

  For more information, see "Importing Functional Testing files in a Guided Scan" on page 135.
- **Export** a recorded macro. After a macro is selected or recorded, you may optionally specify allowed hosts. Opens a standard file-selection window, allowing you to save a recorded macro.

After you specify and play a workflow macro, it appears in the Workflows table and its Allowed Hosts are added to the **Guided Scan > Workflows > Workflows > Manager Workflow** page. You can enable or disable access to particular hosts. For more information, see "Scan settings: Allowed Hosts" on page 375.

Adding Burp Proxy results

If you have run Burp Proxy security tests, the traffic collected during those tests can be imported into a Workflows macro, reducing the time it would otherwise take to rescan the same areas.

To add Burp Proxy results to a workflow macro:

1. If you are not on the Workflows screen, click on the **Manage Workflows** step in the Guided Scan tree.
2. Click the **Import** button.

   The Import Macro file selector appears.
3. Change the file type box filter from **Web Macro (*.webmacro)** to **Burp Proxy (*.*)**.
4. Navigate to your Burp Proxy files and select the desired file.
5. Click **Open**.

Using the Profiler

The OpenText DAST Profiler conducts a preliminary examination of the target website 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 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 Profiler's suggestion to configure the required information before continuing.

Similarly, your settings may specify that OpenText DAST should not conduct "file-not-found" detection. This process is useful for websites 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 OpenText DAST setting to accommodate this feature.

To launch the Profiler:

1. Click **Profile**.
   The Profiler runs. For more information, see "Server Profiler" on page 238.
   Results appear in the Optimize scan for box in the Settings section .
2. If necessary, provide any requested information.
3. Click the **Next** button.

Several options may be presented even if you do not run the Profiler, as described in the following sections.

Autofill web forms

Select **Auto-fill Web forms during crawl** if you want OpenText DAST to submit values for input controls on forms it encounters while scanning the target site. OpenText DAST will extract the values from a prepackaged default file or from a file that you create using the Web Form Editor. See the Web Form Editor chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*. You may:

1. Click the browser button to locate and load a file.
2. Click **Edit** to edit the selected file (or the default values) using the Web Form Editor.
3. 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. For more information, see "Scan settings: Allowed Hosts" on page 375.

To add allowed domains:

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

Reuse identified Suppressed Findings

You can import vulnerabilities that were changed to false positive or ignored in previous scans. If those false positive or ignored items match vulnerabilities detected in the current scan, the vulnerabilities will be changed to false positive or ignored. You can import suppressed findings from existing scans or suppressed findings files. For more information, see "Suppressed findings" on page 67.

To reuse identified suppressed findings:

1. Select **Import Suppressed Findings**.
2. Continue according to the following table.

To use...	Then...
Existing scans	a. Click **Click here to import suppressed findings from scans**. The Select a Scan to Import Suppressed Findings dialog opens. b. Select one or more scans containing suppressed findings from the same site you are now scanning. c. Click **OK**.
Suppressed findings files	a. Click **Click here to import suppressed findings from a file**. A standard Windows file selection dialog box opens. b. Select the file to import, and then click **Open**. c. Optionally, repeat Steps a and b to select additional files.

Apply sample macro

OpenText'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 OpenText DAST and the responses returned by the target server.

While scanning a website, OpenText DAST displays in the navigation pane only those sessions that reveal the hierarchical structure of the website, plus those sessions in which a vulnerability was discovered. However, if you select **Enable Traffic Monitor**, OpenText DAST adds the **Traffic Monitor** button to the Scan Info panel, allowing you to

display and review each HTTP request sent by OpenText DAST and the associated HTTP response received from the server.

Message

If the Profiler does not recommend changes, the Guided Scan wizard displays the message "No settings changes are recommended. Your current scan settings are optimal for this site."

Click **Next**.

The Final Review page appears with **Configure Detailed Options** highlighted in the left pane.

Configuring additional options

To configure detailed options, specify any of the following settings.

Reuse identified False Positives

Select the **False Positives** box to reuse false positives that OpenText DAST has already identified.

Traffic analysis

1. To use the Web Proxy tool, select Launch and Direct Traffic through Web Proxy to use the Web Proxy tool to examine the HTTP requests issued by OpenText DAST and the responses returned by the target server.

   Web Proxy is a stand-alone, self-contained proxy server that you can configure and run on your desktop. Web Proxy enables you to monitor traffic from a scanner, a web browser, or any other tool that submits HTTP requests and receives responses from a server. Web Proxy is a tool for a debugging and penetration scan; you can view every request and server response while browsing a site.
2. Select the **Traffic Monitor** box to display and review each HTTP request sent by OpenText DAST and the associated HTTP response received from the server.

   While scanning a website, OpenText DAST displays only those sessions that reveal the hierarchical structure of the website, plus those sessions in which a vulnerability was discovered. However, if you select **Enable Traffic Monitor**, OpenText DAST enables you to display and review each HTTP request sent by OpenText DAST and the associated HTTP response received from the server.
3. Click **Next**.

   The Validate Settings and Start Scan page appears with **Configure Detailed Options** highlighted in the left pane.

Validating settings and starting the scan

Options on this page allow you to save the current scan settings.

1. To save your scan settings as an XML file, select **Click here to save settings**. Use the standard Save as window to name and save the file.
2. In the **Scan Now** area, review your scan settings, and then click **Start Scan** to begin the scan.

See also

"Guided Scan overview " on page 95

Using the Native Scan template

OpenText DAST allows you to scan the back-end traffic generated by your Android or iOS app or service. Traffic can be generated by running your application on an Android, Windows, or iOS device, or by running the software through an Android or iOS emulator.

The Guided Scan wizard will step you through the necessary stages and steps required to scan your application back-end traffic. If you need to return to a previous step or stage, click the back navigation button, or click the step in the Guided Scan tree to be taken directly there.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Setting up your mobile device

Running a native scan requires that you configure the mobile device to work with a secure proxy. In order to do that, you will need to:

- Set up a Mobile Device/Emulator Proxy (see "Setting the mobile device proxy address" on page 125)
- Install a Trusted Certificate (see "Adding a trusted certificate" on page 126)

Understanding Guided Scan stages

A Guided Scan using a native mobile template consists of four stages, each of which has one or more steps. The stages are:

Native Mobile: where you choose a device or emulator, configure device/emulator proxy, and select the type of scan you want to run.

Login: where you define the type of authentication if back-end of your mobile application requires it.

Application: where you run your app, record web traffic, and identify the hosts and RESTful endpoints to include in your scan.

Settings: where you review and validate your choices and run the scan.

Supported devices

OpenText DAST supports scanning the back-end traffic on Android, Windows, and iOS devices as described in the following table.

OS	Supported Devices
Android	Any Android device, such as an Android-based phone or tablet
Windows	Any Windows device, such as a Windows phone or Surface tablet
iOS	Any iOS device, such as a iPhone or iPad, running the latest version of iOS

Supported development emulators

In addition to support for Android and iOS devices, you can run your application through your Android or iOS emulator in your development environment. When scanning traffic generated via your device emulator, you must ensure that the development machine is on the same network as OpenText DAST and that you have set up a proxy between OpenText DAST and your development machine.

Launching a Native Scan

In order to launch a Native Scan, you will need to make sure your device or emulator is on the same network as OpenText DAST. In addition, you need to have authorization and access to the ports on the machine where you are running OpenText DAST in order to successfully create a proxy connection.

To launch a Native Scan:

1. Open OpenText DAST.
2. Click **Start a Guided Scan** on the OpenText DAST Start page.
3. Select **Native Scan** from the Mobile Templates section.

   The Guided Scan wizard displays the first step in the Native Mobile stage: Choose Device/Emulator.

Choosing device/emulator type

After launching the Guided Scan, you are provided with the options described in the following table.

Option	Description
Profile	The type of device or emulator you want to scan. Select a type from the drop-down menu. For more information, see "Selecting a profile" below.
Mobile Device/Emulator Proxy	The IP address and port number for the proxy that OpenText DAST creates for listening to the traffic between your device or emulator and the web service or application being tested. Unless the IP address and/or port are reserved for other activities, use the default settings. For more information, see "Setting the mobile device proxy address" below.
Trusted Certificate	The port and URL to acquire a client certificate for your device or emulator. To download and install the certificate on your device or emulator, see "Adding a trusted certificate" on the next page.

Selecting a profile

To set the device profile, select one of the following from the **Profile** drop-down textbox:

- iOS Device - An iPad or iPhone running the latest version of iOS.
- iOS Simulator - The iOS emulator that is part of the iOS SDK.
- Android Device - A phone or tablet running the Android operating system.
- Android Emulator - The Android emulator that is part of the Android SDK.
- Windows Device - A Windows phone or Surface tablet.

Setting the mobile device proxy address

The Mobile Device/Emulator Proxy section lists the Host IP address and the Port number that will be used to establish a proxy connection between your device or emulator and OpenText DAST. Use the suggested settings unless the IP address or port number are unavailable on your system.

Note: If you are unable to connect to the server or access the Internet after setting your proxy, you may need to open up or change the port on your firewall specified in the Native Mobile stage. If it still does not work, you may need to select a different IP address. The IP address presented in the OpenText DAST interface enables you to

click the address and select an alternate from a drop-down list.

To set up a proxy on an iOS device:

1. Run the **Settings** application.
2. Select **Wi-Fi**.
3. Select the Wi-Fi network you are using to connect to OpenText DAST.
4. Scroll down to the HTTP Proxy section and select **Manual**.

   The screen displays the network configuration options for the network your device is connected to.
5. Scroll down further and type in the Server IP address and the Port number provided by OpenText DAST. If you don't have this information, see "Choosing device/emulator type " on the previous page.
6. In OpenText DAST, click the **Verify** button in the Trusted Certificate section to verify the connection is working properly.

   The Verify activity progress bar appears.
7. Launch the default browser on your device and visit any site to verify that OpenText DAST is able to see the back-end traffic.

   If everything is configured properly, after a few moments, the Verify activity progress bar will state that the traffic has been successfully verified.
8. Click **OK** to dismiss the verification progress bar and then click **Next** to select a scan type.

To set up a proxy on an Android or Windows device, consult your operator's instructions.

Adding a trusted certificate

If your site requires a secure connection, each time you run a scan, OpenText DAST generates a unique client certificate for your device or emulator. You will need to install the certificate into the device's (or emulator's) certificate repository.

Note: You can add a client certificate to a Windows phone, but the only way to subsequently remove it is to restore the phone to its default settings.

There are three ways to add a certificate:

- Scan the QR code from the Trusted Certificate section of Guided Scan (requires QR reader software).
- Type the address into the built-in browser on your device or device emulator.
- Copy the certificate to your system clipboard for applying later (used when scanning with a device emulator).

Choose the option that best suits your needs.

Note: After completing the scan, you should remove the certificate from the repository on your device. See "Post scan steps" on page 134.

To Add a Certificate to an iOS device or emulator:

1. After scanning the QR code or typing the provided URL into your browser, the Install Profile page appears.

   Note: The OpenText DAST Root certificate status will display as Not Trusted until you add it to your root chain.

2. Tap the **Install** button.

   A warning screen will appear stating that the certificate is not trusted. Once you add the certificate to the certificate repository on your device or emulator, the warning will go away.
3. Tap **Install** on the Warning screen.

   The display changes to that of the current network your device or emulator is connected to. Make sure it is connected to the same network as OpenText DAST.

Choosing the scan type

After setting up your device or emulator to work with OpenText DAST during the first part of the Native Mobile stage, you will need to select the type of scan you would like to run.

Set the options as described in the following table.

Option	Description
Scan Name	Type a name for the scan so that later you can identify the scan on the Manage Scans page.
Scan Method	Choose the type of scan you want from the following list: • **Crawl Only**: maps the attack surface of the specified workflow(s). • **Crawl and Audit**: maps the attack surface of the specified workflow(s) and scans for vulnerabilities. • **Audit Only**: only attack the specified workflows.
Policy	Select a policy for the scan from the drop-down menu. For more information on policies, see "OpenText DAST policies" on page 458. For information on creating and editing policies, see the Policy Manager chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Option	Description
Crawl Coverage	Select the level of coverage you want using the Crawl Coverage slider.

Configuring network authentication

If your network requires user authentication, you can configure it here. If your network does not require user authentication, click the Next navigation button or the next appropriate step in the Guided Scan tree to continue on.

To configure network authentication:

1. Click the **Network Authentication** checkbox.
2. Select a **Method** from the drop-down list of authentication methods. The authentication methods are:
   - **ADFS CBT**
   - **Automatic**
   - **Basic**
   - **Digest**
   - **Kerberos**
   - **Negotiate**
   - **NT LAN Manager (NTLM)**
   - **OAuth 2.0 Bearer**
3. Do one of the following:
   - For all authentication methods *except* OAuth 2.0 Bearer, type a user ID in the **User Name** box and the user's password in the **Password** box.
   - For the OAuth 2.0 Bearer method, click **Configure** and continue with "Configuring OAuth 2.0 bearer credentials" on page 400.

Using a client certificate

To use a client certificate for network authentication: :

1. Select the **Client Certificate** check box.
2. Do one of the following:
   - To use a certificate that is local to the computer and is global to all users on the computer, select **Local Machine**.

- To use a certificate that is local to a user account on the computer, select **Current User**.

  Note: Certificates used by a common access card (CAC) reader are user certificates and are stored under Current User.

3. Do one of the following:
   - To select a certificate from the "Personal" ("My") certificate store, select **My** from the drop-down list.
   - To select a trusted root certificate, select **Root** from the drop-down list.
4. Does the website use a common access card (CAC) reader or a certificate that is password protected?
   - If *yes*, do the following:
     i. Select a certificate that is prefixed with “(Protected)” from the **Certificate** list.

     Information about the selected certificate and a Password/PIN field appear in the Certificate Information area.

     ii. If a password or PIN is required, type it in the **Password/PIN** field.

     Note: If a PIN is required and you do not enter the PIN at this point, you must enter the PIN in the Windows Security window each time it prompts you for it during the scan.

     Important! By default, OpenText DAST uses OpenSSL. If you are using a specific SSL/TLS protocol rather than OpenSSL, the Profiler portion of scan configuration may not work with certificates that are protected with a password.

     iii. Click **Test**.

     If you entered the correct password or PIN, a Success message appears.
   - If *no*, select a certificate from the **Certificate** list.

     Information about the selected certificate appears below the Certificate list.

Configuring application authentication

If your site requires authentication, you can use this step to create, select, or edit a login macro to automate the login process and increase the coverage of your site. A login macro is a recording of the activity that is required to access and log in to your application, typically by entering a user name and password and clicking a button such as Log In or Log On.

If **Enable macro validation** is selected in Scan Settings: Authentication for scans that use a login macro, OpenText DAST tests the login macro at the start of the scan to ensure that the log in is successful. If the macro is invalid and fails to log in to the

application, the scan stops and an error message is written in the scan log file. For more information and troubleshooting tips, see "Testing login macros" on page 494.

Note: Macro testing is not supported for macros containing two-factor authentication.

Important! If you use a macro that includes Two-factor Authentication, then you must configure the Two-factor Authentication Application settings before starting the scan. For more information, see "Application settings: Two-Factor Authentication" on page 442.

The following options are available for login macros:

- "Using a login macro without privilege escalation" below
- "Using login macros for privilege escalation" below
- "Testing the macro" on page 132

Masked values supported

If the macro uses parameters for which values are masked in the Web Macro Recorder, then these values are also masked when configuring a Guided Scan in OpenText DAST.

Using a login macro without privilege escalation

To use a login macro:

1. Select the **Use a login macro for this site** check box.
2. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
3. Click the **Next** button.

   The Application Authentication Step is complete. Proceed to the Application Stage to run your application.

Using login macros for privilege escalation

If you selected the Privilege Escalation policy or another policy that includes enabled Privilege Escalation checks, at least one login macro for a high-privilege user account is required. For more information, see "About privilege escalation scans" on page 198. To use login macros:

1. Select the **High-Privilege User Account Login Macro** check box. This login macro is for the higher-privilege user account, such as a Site Administrator or Moderator account.
2. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

   After recording or selecting the first macro and clicking the next arrow, a "Configure Low Privilege Login Macro" prompt appears.
3. Do one of the following:
   - To perform the scan in authenticated mode, click **Yes**. For more information, see "About privilege escalation scans" on page 198.

     Guided Scan returns to the Select Login Macro window for you to create or select a low-privilege login macro. Continue to Step 4.
   - To perform the scan in unauthenticated mode, click **No**. For more information, see "About privilege escalation scans" on page 198.

     The Application Authentication Step is complete. Proceed to the Application Stage.
4. Select the **Low-Privilege User Account Login Macro** check box. This login macro is for the lower-privilege user account, such as a viewer or consumer of the site content.
5. Do one of the following:
   - To use a pre-recorded login macro, click the ellipsis button (**...**) to browse for a saved macro.
   - To edit an existing login macro shown in the Login Macro field, click **Edit**.
   - To record a new macro, click **Create**.

   For details about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
6. After recording or selecting the second macro, click the **Next** button.

   The Application Authentication Step is complete. Proceed to the Application Stage to run your application.

Testing the macro

Optionally, click **Test** to locate the login form and run macro validation tests before advancing to the next stage in the Guided Scan wizard. If you need to cancel the validation test prior to completion, click **Cancel**.

If the macro is invalid and fails to log in to the application, an error message appears. For more information and troubleshooting tips, see "Testing login macros" on page 494.

Running the application

To run the application and generate and collect web traffic:

1. Click the **Record** button.
2. Exercise the application, navigating through the interface as your customers will.
3. When you have generated enough traffic, click the **Stop** button.
4. Click **Play** to verify your workflow.

Finalizing allowed hosts and RESTful endpoints

After running the application and collecting web traffic, a list will be generated of the Allowed Hosts and potential RESTful Endpoints.

To select the hosts to include in your audit, click the check boxes in the **Enabled** column of the Allowed Hosts table.

The list of RESTful endpoints is generated by listing every possible combination that could be a RESTful endpoint. Select the actual RESTful endpoints from the list by selecting their Enabled check boxes. To reduce the list to a more likely subset, click the Detect button. Heuristics are applied, filtering out some of the less likely results. Select the Enabled check boxes from the resultant list.

If OpenText DAST didn't find all of the RESTful endpoints, you can add them manually.

To set up a new RESTful endpoint rule:

1. Click the **New Rule** button.
   A new rule input box appears in the RESTful Endpoints table.
2. Following the sample format in the input box, type in a RESTful Endpoint.

To Import a List of RESTful Endpoints:

1. Click the **Import** button.
   A file selector appears.
2. Select a Web Application Description Language (.wadl) file.
3. Click **OK**.

Reviewing settings

During the final stage, you can set a number of options that affect how the collected traffic is audited. The available options vary, based on the selections you have made.

Configure detailed options

The Configure Detailed Options step enables you to set detailed options. These options will change from scan to scan, as they are dependent on the choices made in the Guided Scan wizard. Some of the options include:

Reuse identified suppressed findings

You can import vulnerabilities that were changed to false positive or ignored in previous scans. If those false positive or ignored items match vulnerabilities detected in the current scan, the vulnerabilities will be changed to false positive or ignored. You can import suppressed findings from existing scans or suppressed findings files. For more information, see "Suppressed findings" on page 67.

To reuse identified suppressed findings:

1. Select **Import Suppressed Findings**.
2. Continue according to the following table.

To use...	Then...
Existing scans	a. Click **Click here to import suppressed findings from scans**. The Select a Scan to Import Suppressed Findings dialog opens. b. Select one or more scans containing suppressed findings from the same site you are now scanning. c. Click **OK**.
Suppressed findings files	a. Click **Click here to import suppressed findings from a file**. A standard Windows file selection dialog box opens. b. Select the file to import, and then click **Open**. c. Optionally, repeat Steps a and b to select additional files.

Traffic analysis

You can use a self-contained proxy server on your desktop. With it you can monitor traffic from a scanner, a browser, or any other tool that submits HTTP requests and received responses from a server. You can also enable the Traffic Monitor and display the hierarchical structure of the website or web service in an OpenText DAST navigation pane. It enables you to display and review every HTTP request sent by OpenText DAST and the associated HTTP response received from the server.

Scan mode

A crawl-only feature that enables you to set Discovery (Path Truncation). Path truncation allows you to make requests for known directories without file names. This can cause directory listings to be displayed. You can also select the Passive Analysis (Keyword Search) option to examine every response from the web server for (error messages, directory listings, credit card numbers, etc.) not properly protected by the website.

Validating settings and starting the scan

Options on this page allow you to save the current scan settings.

1. To save your scan settings as an XML file, select **Click here to save settings**. Use the standard Save as window to name and save the file.
2. In the **Scan Now** area, review your scan settings, and then click **Start Scan** to begin the scan.

Post scan steps

After you have completed your scan and run OpenText DAST, you will need to reset your Android, Windows, or iOS device or emulator to its former state. The following steps show how to reset your iOS device to the way it was before you began. Steps for other devices and emulators are similar, but depend on the version of the OS you are running.

To remove the Fortify Certificate on an iOS device:

Run the Settings application.

1. Select **General** from the Settings column.
2. Scroll down to the bottom of the list and select **Profile WebInspect Root**.
3. Tap the **Remove** button.

To Remove the Proxy Settings on an iOS device:

1. Run the **Settings** application.
2. Select **Wi-Fi** from the **Settings** column.
3. Tap the **Network** name.

Delete the Server IP address and the Port number.

See also

"Guided Scan overview " on page 95

Importing Functional Testing files in a Guided Scan

If you have the OpenText™ Functional Testing application installed, OpenText DAST detects it and enables you to import a functional testing file (.usr) into your workflow scan to enhance the thoroughness and attack surface of your scan. For more information, see Functional Testing on the OpenText website.

To import a functional testing (`.usr`) file into an OpenText DAST Guided Scan:

1. Launch a Guided Scan, and then select Workflows Scan as the Scan Type. Additional text appears under the Workflows scan option:

   OpenText Functional Testing has been detected. You can import scripts to improve the thoroughness of your security test.
2. Click the **Next** button.
3. In the Authentication section, **Application Authentication** is automatically selected. Complete the fields as indicated.
4. On the Manage Workflows screen, click **Import**. The Import Scripts dialog box appears. On the Import Scripts dialog box, you may:
   - Type the filename.
   - Browse to your file by clicking to locate your file with a `.usr` extension. Select **VuGen script file** from the drop-down file type, and then navigate to the file.
   - Click **Edit** to launch the OpenText Functional Testing application.
5. (Optional) On the Import Scripts dialog box, you may select either of the following options:
   - **Show OpenText Functional Testing UI during import**
   - **Open script result after import**
6. Select the file to import, and then click **Import**. After your file is successfully imported, the file appears in the Workflows table.
7. Select one of the following from the Workflows table:
   - **Record** - launches the Web Macro Recorder. For more information, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
   - **Edit** - enables you to modify the file using the Web Macro Recorder. See the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
   - **Delete** - deletes the script from the Workflows table.

- **Import** - imports another file.
- **Export** - saves a file in .webmacro format with the name and location you specify

8. Click the **Next** button.

   When the first `.usr` script file is added to the list, its name (or default name) appears in the Workflows table and an Allowed Hosts table is added to the pane.

   Adding another `.usr` script file can add more allowed hosts. Any host that is enabled is available to all the listed workflow `.usr` script files, not just the `workflow.usr` file for which it was added. The Guided Scan will play all the listed workflow files and make requests to all the listed allowed hosts, whether or not their check boxes are selected. If a check box for an allowed host is selected, OpenText DAST will crawl or audit the responses from that host. If a check box is not selected, OpenText DAST will not crawl or audit the responses from that host. In addition, if a particular workflow `.usr` script uses parameters, a Macro Parameters table is displayed when that workflow macro is selected in the list. Edit the values of the parameters as needed.
9. After you have completed changes or additions to the Workflows table, proceed in the Guided Scan wizard to complete your settings and run the scan. For more information about recording a new login macro or using an existing login macro, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Running an API or web service scan

You can run a scan of the REST application programming interface (API) or web service used in your target applications. OpenText DAST supports conducting scans of the following API or web service technologies:

- GraphQL
- gRPC
- OData
- Postman
- SOAP
- Swagger (also known as Open API)

Important information about SOAP web service scans

The legacy SOAP web service scan functionality will be removed in a future release. OpenText recommends transitioning to the new API scan for SOAP as soon as possible.

Important information about gRPC proto files

All gRPC proto files must be self-contained. Any imports must be to internally recognized resources and not to user-generated files. OpenText DAST cannot identify file paths from imported proto files. If such files are used, the scan will fail to generate the client and will be interrupted. If additional imports are needed, they must be combined with the primary proto file into a "master" proto file.

Known limitations of gRPC scans

Be aware of the following known limitations associated with gRPC scans:

- OpenText DAST installed on Windows 11 or a Linux version of OpenText DAST is required for conducting scans of gRPC APIs.
- You must use a Linux version of OpenText DAST to conduct a scan of a gRPC API running on a server with unencrypted HTTP/2 (H2C).

For more information about the Linux versions, see *OpenText™ Dynamic Application Security Testing and OAST on Docker User Guide*.

Options for conducting scans

You can conduct API and web service scans by way of the OpenText DAST user interface or the CLI. For more information, see the following topics:

- "Using the API Scan Wizard" below
- "Scanning an API with wi.exe" on page 155

For OData and Swagger (Open API) types, you may use the `WISwag.exe` tool in conjunction with wi.exe or the OpenText DAST REST API to conduct a scan. For more information, see "Using the WISwag.exe tool" on page 305.

Using the API Scan Wizard

You can use the API Scan Wizard to configure settings for an API scan or a web service scan in the OpenText DAST user interface.

API scans

For Swagger, OData, and Postman scans, OpenText DAST creates a macro from the REST API definition, and then performs an automated analysis. For GraphQL, gRPC, and SOAP scans, a more traditional scanning method is used.

Important! If you are configuring a Postman API scan, be sure that the prerequisite software is installed before proceeding. For more information about this and other

aspects of using Postman collection files, including configuring dynamic authentication using dynamic tokens, see "Scanning with a Postman collection" on page 319.

Web service scans

For a legacy web service scan, OpenText DAST crawls the WSDL site and submits a value for each parameter in each operation it discovers. OpenText DAST then audits the site by attacking each parameter in an attempt to detect vulnerabilities such as SQL injection.

See "Auditing web services" on page 247 for more information on how a web services vulnerability scan differs from other types of scan actions.

Note: As of version 26.4, OpenText DAST, legacy WSDL web service scans are no longer available. Web service scans that were run with a previous version, and that are saved in OpenText DAST, can be opened but can no longer be rerun.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Getting started with the API Scan Wizard

To begin configuring settings for an API scan or a web service scan:

1. On the OpenText DAST **Start Page**, click **Start an API Scan**.
   The API Scan Wizard opens.
2. Optionally, enter a name for the scan in the **Scan Name** box.

Tip: On any window presented by the API 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 that 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 OpenText DAST **Edit** menu, and then select **Default Scan Settings**.

What's next?

Do one of the following:

- To configure an API scan, proceed with "Configuring an API scan" on the next page.
- To configure a legacy web services scan using a Web Service Definition Language (WSDL) file, proceed with "Using the API Scan Wizard" on the previous page.

Configuring an API scan

You can begin configuring settings for an API scan in the **API Scan** page of the **API Scan Wizard**.

To configure settings for an API scan:

1. Select **API Scan**.
2. In the **API Type** list, select the API type to be scanned. The options are:
   - **GraphQL**
   - **gRPC**

     Note: OpenText DAST on Windows using a SOCKS proxy cannot scan gRPC APIs.
   - **OData**
   - **Postman**
   - **SOAP**
   - **Swagger** (also known as Open API)
3. Continue according to the following table.

For this API type...	Do this...
GraphQL **gRPC** **OData** **Swagger**	Do one of the following: • In the **API Definition/Config** box, provide the URL to the API definition file, as shown in the following examples: `http://172.16.81.36/v1` `http://myapi/protos/client.proto` `http://myapi/graphql/` • Click ... and import a configuration file or definition file. **Tip:** Alternatively, you can paste the full path to the file that is saved on your local machine. If you did not enter a name in the **Scan Name** box, the definition file is parsed and the URL is added to the **Scan Name** box.
Postman	Do one of the following: • To import a workflow collection, click ..., select **Workflow** from the drop-down list, and then import the Postman workflow

For this API type...	Do this...
	collection file. • To import an authentication collection, click ..., select **Authentication** from the drop-down list, and then import the Postman authentication collection file. • To import an environment file, click ..., select **Environment** from the drop-down list, and then import the Postman environment file. • To import a file that includes global variables, click ..., select **Globals** from the drop-down list, and then import the Postman global variables file. The file is added to the list of Postman files. Repeat this Step to import additional files. **Important!** You can import multiple workflow collection files. You can import only one authentication collection file, one environment file, and one global variables file. If the same variables are defined in both the environment file and the global variables file, then the environment file overrides the global file.
SOAP	a. Do one of the following: ◦ In the **API Definition/Config** box, provide the URL to the API definition file, as shown in the following example: `http://172.16.81.36/web-services/infoService?wsdl` ◦ Click ... and import a configuration file or definition file. **Tip:** Alternatively, you can paste the full path to the file that is saved on your local machine. If you did not enter a name in the **Scan Name** box, the definition file is parsed and the URL is added to the **Scan Name** box. b. In the **Version** list, select a version to allow filtering of operations by the specific version. Options are as follows: ◦ **Legacy** – filters against the lowest supported version.

For this API type...	Do this...
	◦ **Mixed** – uses a combination of Legacy and Newest, depending on what is available. ◦ **Newest** – the default setting, filters against the latest version.

4. If you imported a definition file or a configuration file in which a scheme, host, or service path is specified, the **API location is different from API definition location** option is selected. Specify the following:
   - In the **API Scheme Type** list, select a type. Options are **HTTP**, **HTTPS**, and **HTTP and HTTPS**.
   - In the **API Host** box, type the URL or hostname.
   - In the **API Service Path** box, type the directory path for the API service.

   Note: The GraphQL service location is always the same as the definition location. The gRPC service location is always different from the definition location.

   Note: For SOAP, if the query string "?wsdl" value is removed, then the SOAP service location may or may not be the same as the definition location. Additionally, if you provide a WSDL with no defined service host or path, you must select **API location is different from API definition location** and leave the **API Host** and **API Service Path** boxes empty. OpenText DAST will use the SOAP port binding locations for sending requests.

   Note: If the service path is not defined for a Swagger scan, then OpenText DAST will use the basePath that is defined in the Swagger definition contents. For Swagger scans, select **API location is different from API definition location** unless your service is explicitly run at the same location as the docs folder for Swagger. Optionally, you may choose to define a service path if it differs from the basePath.

5. Click **Next** and proceed with "Configuring authentication and connectivity for API scans" on the next page.

Configuring authentication and connectivity for API scans

You can configure proxy settings, network authentication, and site authentication on the **Authentication and Connectivity** page of the **API Scan Wizard**. Options for configuring authentication include the following:

- "Configuring proxy settings for API and web service scans" below
- "Configuring network authentication for API and web service scans" on the next page
- "Using a client certificate" on page 145
- "Using custom headers" on page 146
- "Configuring SOAP authentication" on page 147

Note: Some options in this topic do not apply to legacy web services scans using a Web Service Definition Language (WSDL) file or an existing Web Service Test Design (WSD) file.

Configuring proxy settings for API and web service scans

If you need to access the target site through a proxy server, you can configure proxy settings on the **Authentication and Connectivity** page of the **API Scan Wizard**.

To configure proxy settings:

- Select **Network Proxy** and then choose an option from the **Proxy Profile** list:
  - **Auto Detect**: Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's Web proxy settings.
  - **Use System Proxy**: Import your proxy server information from the local machine.
  - **Use PAC File**: Load proxy settings from a Proxy Automatic Configuration (PAC) file. 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.

Important! Socks4 proxy servers do not support authentication. When using a Socks proxy server that requires authentication, you must use a Socks5 proxy.

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.

Configuring network authentication for API and web service scans

You can configure network authentication for accessing the Web server on the **Authentication and Connectivity** page of the **API Scan Wizard**.

To configure network authentication for the Web server:

1. Select **Network Authentication**.
2. In the **Method** drop-down list, select an authentication method. The API Type determines the available authentication methods. The complete list of methods is:
   - **ADFS CBT**
   - **Automatic**
   - **Basic**
   - **Bearer**
   - **Custom**
   - **Digest**
   - **Kerberos**
   - **Negotiate**
   - **NT LAN Manager (NTLM)**
   - **OAuth 2.0 Bearer**

   Note: The ADFS CBT, Automatic, Kerberos, and Negotiate methods are not applicable to scans that use AuthProviders.
3. Continue according to the following table.

For this authentication type...	Do this...
ADFS CBT **Automatic** **Basic** **Digest** **Kerberos** **Negotiate** **NTLM**	a. Type the authentication username in the **Username** box. b. Type the authentication password in the **Password** box.
Custom	a. Type the custom header name or token name in the

For this authentication type...	Do this...
	Scheme box. b. Type the token value in the **Parameter** box. When using Custom, you can fetch a token that is generated from a response to a workflow macro, and then use the token to apply state. For more information, see "Fetching a token value" below. **Note:** Not available for SOAP web service scans.
Bearer	Type the token value in the **Parameter** box. When using Bearer, you can fetch a token that is generated from a response to a workflow macro, and then use the token to apply state. For more information, see "Fetching a token value" below. **Note:** Not available for SOAP web service scans.
OAuth 2.0 Bearer	Click **Configure** and continue with "Configuring OAuth 2.0 bearer credentials" on page 400.

Fetching a token value

You can use a custom regular expression to fetch the token value from a login or workflow macro. If a match to the regular expression occurs in the response, then the value is fetched and used as a bearer token. If the regular expression contains parentheses, then the value inside the parentheses will be extracted and used as a bearer token. Only the first value inside parentheses will be used.

Note: Fetching a token value does not apply to OData or Swagger definition types.

To fetch a token value:

1. Select **Fetch Token From Macro**.
2. Do one of the following:
   - To import an existing macro, click ..., and then locate and select the file to import.
   - To record a macro, click Record....
3. Type a regular expression for pattern matching in the **Fetch Token Search Pattern** box.

4. Do one of the following:
   - To have each scan thread run its own fetch macro playback and apply the bearer token value to the thread, select the **Isolate State** check box.
   - To have only one fetch macro playback run for all scan threads and the single shared bearer token value apply to all threads, clear the **Isolate State** check box.

Using a client certificate

Client certificate authentication allows users to present client certificates rather than entering a user name and password for site authentication. You can enable the use of a certificate and then import the certificate to the scan settings.

Note: Client certificates do not apply to OData or Swagger definition types.

To use a client certificate:

1. Select **Client Certificate**.
2. Click ... .
   A standard Windows file selection dialog box opens.
3. Locate and select the certificate file, and then click **Open**.
   The certificate file is added to the Client Certificate box.
4. Enter the password in the **Client Certificate Password** box.

Updating certificates in composite scan settings

The composite scan settings include a BIN file that contains encrypted certificate data. If you need to replace or update the client certificate in your composite scan settings, you can place the updated PFX or P12 file inside the certificates directory in the composite settings ZIP file. When OpenText DAST opens the settings, it will check for PFX and P12 files first. If none are present, then the BIN file will be decrypted and used. For more information about composite settings, see "Application settings: General" on page 431.

To replace or update the client certificate:

1. Locate the encrypted BIN file in the certificates directory in your composite scan settings ZIP. The file name is a GUID, similar to the following:
   `<your-scansettings.zip>\certificates\0b627638-efda-4d01-a83e-80ee3a79b4cf.bin`

   Note: The default settings file location in Windows is `C:\ProgramData\HP\HP WebInspect\Settings\`.
2. Place the updated PFX or P12 file into the same directory.
3. Rename the PFX or P12 file the same name as the BIN file. Using the previous example, the file name will be as follows:
   `0b627638-efda-4d01-a83e-80ee3a79b4cf.pfx`

– OR –

`0b627638-efda-4d01-a83e-80ee3a79b4cf.p12`

Important! Be sure to retain the original file extension.

4. Optionally, if you want to retain an encrypted certificate in the settings, save the settings again and the BIN file will reflect the updated PFX or P12 certificate. The PFX or P12 certificates will be removed from the ZIP.

Tip: PFX and P12 certificates frequently require a password. Use one of the following options to provide the password for the settings:

- Set an empty password when you create the PFX or P12 certificate and place it in the settings ZIP file.
- Keep the password for the PFX or P12 certificate and edit the `settings.json` file to place the password as the value for the CertificatePin as follows:

  `"CertificatePin": "<password>"`

Using custom headers

If additional or different headers are required for authentication purposes, then you must add the information as a Custom Header.

You can configure multiple custom headers.

Important! You cannot configure more than one custom header using the same HTTP header name.

To add a custom header:

1. Select **Custom Headers**.
2. Click **Add...**.
3. In the **Name** box, type the custom HTTP header name. For example, `X-MyCustomAuth`.

   Important! The header must be unique and cannot be `Authorization`.
4. In the **Scheme** box, type the header value prefix name. For example, `CustomToken`.
5. In the **Parameter** box, type the custom header value.
6. Click **OK**.

   The custom header is added to the list.

To edit a custom header:

1. In the **Custom Headers** list, select the custom header you want to edit.
2. Click **Edit...**.
3. Follow steps 3 through 6 of the procedure "To add a custom header:" above.

To remove a custom header:

1. In the **Custom Headers** list, select the custom header you want to remove.
2. Click **Remove**.

Configuring SOAP authentication

You can configure message-based authentication for SOAP scans.

To configure SOAP authentication settings:

1. Select **SOAP Authentication**.
2. Select that authentication method to use from the **SOAP Method** list. Options are **Username Token** and **Certificate Pair**.
3. Continue according to the following table.

For this SOAP method...	Do this...
Username Token	a. In the **Username** box, type the user name whose credentials are used to access the SOAP service. b. In the **Password** box, type the password for the user name. c. In the **Username Token Type** list, select the type of token. Options are **Text** and **Hash**. d. In the **Timestamp** list, select an option for when the Username Token was created and when it expires. Options are **Created**, **Full**, and **None**. e. If nonce is enabled for the token, select **Include nonce**. **Important!** Nonce is required for hash tokens because it helps the server to recalculate the hash and compare it to the data the client sent.
Certificate Pair	a. Click ... to the right of the **Client Certificate** box. A standard Windows file selection dialog box opens. b. Locate and select the certificate file, and then click **Open**. The certificate file is added to the Client Certificate box. c. In the **Client Certificate Password** box, type the password. d. Click ... to the right of the **Server Certificate** box. A standard Windows file selection dialog box opens. e. Locate and select the certificate file, and then click **Open**.

For this SOAP method...	Do this...
	The certificate file is added to the Server Certificate box. f. In the **Server Certificate Password** box, type the password.

4. Optionally, to identify the Web Services Addressing (WS-Addressing) schema version used by the SOAP service, select **WS Addressing** and continue as follows:
   a. In the **Schema Version** list, select the version. Options are **NONE**, **WSA0408**, and **WSA0508**.
   b. In the **WSA: To** box, enter the URL override for the web service host.

      Note: SOAP services may be exposed by way of a load balancer or reverse proxy. This configuration may prevent the sensor from getting the correct information for the internal web service host name. The "WSA: To" URL override provides the correct address into WS Addressing.

      The URL override uses the following format:

      `https://<host_name><service_path>/<port_name>`

What's next?

Do one of the following:

- If you are configuring a legacy web services scan using a Web Service Definition Language (WSDL) file or an existing Web Service Test Design (WSD) file, click **Next** and proceed with Configuring Scan Details for API and Web Service Scans.
- For all other API scans, click **Next** and proceed with "Configuring API content and filters" below.

Configuring API content and filters

When configuring API scans, you can use the **Content and Filters** page of the **API Scan Wizard** to configure the preferred content type, as well as operations and parameter names and types to include or exclude during the scan. If you are conducting a Postman API scan, the scan wizard validates the collection file(s) that you previously selected and displays the Postman configuration settings in this page. You can review the settings and make adjustments as needed.

Viewing and adjusting Postman configuration settings

Note: Postman configuration settings are available only when conducting a Postman API scan.

Upon successful validation of the Postman collection file(s), a list of sessions contained in the collection file(s) appears in the Postman Configuration area. If authentication sessions are identified, they are preselected as **Auth** sessions. All other sessions are preselected as **Audit** sessions. Additionally, the type of authentication detected is listed as the **Token Strategy** with the options of **None**, **Static**, or **Dynamic**.

Note: Auth sessions will be used for authentication for the scan. **Audit** sessions will be audited in the scan.

Optionally, to adjust the settings:

1. Select the **Auth** or **Audit** check box for a session to change its type as needed.
2. Make changes to the Postman authentication settings as follows:
   - For **Static** authentication, enter a token in the **Custom Header Token** box.
   - For **Dynamic** authentication, do the following:
     - Select the **Regex (Custom)** option to the right of the **Response Token** box, and then enter a custom regular expression in the **Response Token Name** box.
     - Select the **Regex (Custom)** option to the right of the **Request Token Name** box, and then enter a custom regular expression in the **Request Token Name** box.
     - Clear the **Use Auto Detect** option to the right of the **Logout Condition** box, and then enter a new logout condition string in the **Logout Condition** box.

     For more information about dynamic authentication for Postman, see "Manually configuring Postman login for dynamic tokens" on page 323.

Important! If you make changes to the Postman authentication settings, they will not be validated unless you return to the **API Scan** page of the **API Scan Wizard**, and then click **Next** again.

Specifying the preferred content type

The preferred content type setting specifies the preferred content type of the request payload. If the preferred content type is in the list of supported content types for an operation, then the generated request payload will be of that type. Otherwise, the first content type listed in an operation will be used. An example of preferred content type is `application/json`.

Important! The **Preferred Content Type** setting does not work with schema-based APIs, such as GraphQL, SOAP, gRPC, and Postman.

To specify the preferred type:

- Type the preferred content type in the **Preferred Content Type** box.

Defining specific operations to include

The Include feature defines an allow list of operation IDs that should be included in the output.

To define a specific operation to include:

1. Select **Specific Operations**.
2. Select **Include**.
3. Click **Add**.
   The Specify Operation dialog box opens.
4. In the **Operation** box, type the operation ID.
5. Click **OK**.
   The operation ID is added to the allow list.

Defining specific operations to exclude

The Exclude feature defines a deny list of operation IDs that should be excluded from the output.

To define a specific operation to exclude:

1. Select **Specific Operations**.
2. Select **Exclude**.
3. Click **Add**.
   The Specify Operation dialog box opens.
4. In the **Operation** box, type the operation ID.
5. Click **OK**.
   The operation ID is added to the deny list.

Editing specific operations

To edit a specific operation in the allow or deny list:

1. Do one of the following:
   - To edit an operation in the allow list, select **Include**.
   - To edit an operation in the deny list, select **Exclude**.
2. Select the operation ID you want to edit.
3. Click **Edit**.

Removing specific operations

To remove a specific operation from the allow or deny list:

1. Do one of the following:
   - To remove an operation in the allow list, select **Include**.
   - To remove an operation in the deny list, select **Exclude**.
2. Select the operation ID you want to remove.
3. Click **Remove**.

Defining parameter rules

Parameter rules define a default value to use for a parameter when the parameter name and type are encountered. You can also specify operations to determine whether a specific parameter rule should or should not apply to those operations.

Important! If you configure a parameter rule and then change the API definition type for which the parameter rule type becomes invalid, the invalid parameter rule type will be changed to **Any**, and a warning message will be displayed below the list.

To add a parameter rule:

1. Select **Parameter Rules**.
2. Click **Add**.
   The Parameter Rule dialog box opens.
3. In the **Parameter Rule Name** box, type a name for the rule.
4. In the **Parameter Rule Type** list, select a type. Available options depend on the API type and may include the following:
   - **Any**
   - **Boolean**
   - **Date**
   - **File**
   - **Guid**
   - **Number**
   - **String**

   For more information on the Parameter Rule Types and their equivalents based on API type, see "Understanding parameter type matches" on page 153.
5. Continue according to the following table:

For this Rule Type...	In the Parameter Rule Value box...
Any	Type any value.
Boolean	Type **true** or **false**.
Date	To enter a string value: • Type a date and time string using the MM/DD/YYYY format as shown in the following example: `04/02/2026 11:00 AM` To select a date and time using a calendar: a. Click the calendar icon ( ). b. Select a date and time. c. Click **Close**.
File	a. Click ..., and browse to locate the file to add to the scan settings. b. Click **Open**.
Guid	Enter a GUID.
Number	Enter a numerical value.
String	Type any value.

6. For OData and Swagger (Open API) scans, in the **Parameter Rule Location** list, select a location where the parameter is found in the request. Options are:
   - **Any**
   - **Body**
   - **Header**
   - **Path**
   - **Query**
7. Optionally, to specify operations to which this parameter rule should or should not apply, select **Specific Operations** and perform steps 2-5 of "Defining specific operations to include" on page 150 or "Defining specific operations to exclude" on page 150.
8. Optionally, select **Inject Parameter** to include the defined parameter in the request.

Important! The **Inject Parameter** option does not work with schema-based APIs, such as SOAP, gRPC, and Postman. Those API types do not accept forced parameters. For GraphQL, **Inject Parameter** only works with the query operation

if the property is in the query schema.

9. Click **OK**.
   The rule is added to the Parameter Rules list.

Editing a parameter rule

To edit a rule in the Parameter Rules list:

- Select the check box for the rule to edit, and then click **Edit**.
  The Parameter Rule dialog box opens. For more information about using this dialog box, see "Defining parameter rules" on page 151.

Removing a parameter rule

To remove a rule from the Parameter Rules list:

- Select the check box for the rule to remove, and then click **Remove**.

What's next?

To configure audit coverage and thoroughness, click **Next** and proceed with "Configuring API audit coverage and thoroughness" on the next page.

Understanding parameter type matches

The following table describes the parameter rule type equivalents by API type.

OpenText DAST Parameter Rule Type	Equivalent				
	Open API (Swagger)	OData	GraphQL	gRPC	SOAP
Any	All	All	All	All	All
Boolean	boolean	Edm.Boolean	boolean	bool	boolean
Date	date (Open API 2.0) string (Open API 3.0)[1]	Edm.Date Edm.DateTime Edm.DateTimeOffset Edm.Duration Edm.Time Edm.TimeOfDay	N/A	N/A	date
File	file (Open API 2.0)[2]	Edm.Binary	N/A	bytes	N/A

[1] Open API 3.0 implementation is qualified by date string format.
[2]Open API 3.0 implementation is qualified by binary or byte string formats.

OpenText DAST Parameter Rule Type	Equivalent				
	Open API (Swagger)	OData	GraphQL	gRPC	SOAP
GUID	N/A	Edm.Guid	N/A	N/A	N/A
Number	number integer	Edm.Byte Edm.Decimal Edm.Double Edm.Int16 Edm.Int32 Edm.Int64 Edm.SByte Edm.Single	int float	double enum fixed32 fixed64 float int32 int64 sfixed32 sfixed64 sint32 sint64 uint32 uint64	base64Binary byte decimal double float hexBinary hexint int integer long signedInt short unsignedByte unsignedInt unsignedLong unsignedShort
String	string	Edm.GeographyCollection Edm.GeographyLineString Edm.GeographyMultiLineString Edm.GeographyMultiPoint Edm.GeographyMultiPolygon Edm.GeographyPoint Edm.GeographyPolygon Edm.GeometryCollection Edm.GeometryLineString Edm.GeometryMultiLineString Edm.GeometryMultiPoint Edm.GeometryMultiPolygon Edm.GeometryPoint Edm.GeometryPolygon Edm.String	id string	string	string

Configuring API audit coverage and thoroughness

The default policy for API and web service scans is the API policy. You can select a different policy or choose additional policies for the scan in the **Audit Coverage and Thoroughness** page of the **API Scan Wizard**. If you select multiple policies, the sensor will aggregate the policies during the scan.

Selecting one or more policies for API scans

To select a different policy:

1. In the **Audit Depth (Policy)** list, slide the toggle for the **API** policy to the disabled position.

2. Slide the toggle for the desired policy to the enabled position.

   The selected policy appears in the ENABLED SCAN POLICIES list. For descriptions of policies, see "OpenText DAST policies" on page 458.
3. Click **Next**.

To select additional policies:

1. In the **Audit Depth (Policy)** list, slide the toggle for the desired policies to the enabled position.

   The selected policies appear in the ENABLED SCAN POLICIES list. For descriptions of policies, see "OpenText DAST policies" on page 458.
2. Click **Next**.

What's next?

To configure scan details, click **Next** and proceed with Configuring scan details for API and web service scans.

Saving settings or starting the API scan

On the **Congratulations** page of the **API Scan Wizard**, you can save the settings you configured or use them to conduct a scan.

Saving settings

If you anticipate running this scan again, you can save the settings in an XML file.

To save the settings:

- Click the **Save** hyperlink to name and save the file.

  When starting a scan through the API Scan Wizard, you can click **Settings** (at the bottom of the window) to load this settings file.

Starting a scan

To start a scan with the settings you configured:

- Click **Scan**.

Scanning an API with wi.exe

You can scan the following API types from the command-line interface (CLI) using wi.exe:

- GraphQL
- gRPC

Note: OpenText DAST on Windows using a SOCKS proxy cannot scan gRPC APIs.

- OData
- SOAP
- Swagger

In the command, you can point to a definition file or an endpoint for the service. Optionally, you can create a scan configuration file that includes additional information, such as authentication and proxy settings, and point to the settings file in the command.

Process overview

The following table describes the process for scanning an API with wi.exe.

Stage	Description
1.	Optionally, prepare an API scan configuration file (JSON). For more information, see "Understanding the API scan configuration file" on the next page. **Tip:** If you do not need any custom configuration, such as authentication, a proxy, or a service path that is different than the URL, then you do not need to create a scan configuration file.
2.	Open the CLI and run a scan using `wi.exe` with the `-api` option as shown in the following examples. `wi.exe -xd -api SOAP -u "D:\Development\soapConfig.json"` `wi.exe -api GraphQL -u "http://localhost:5013/graphql" -tm -pc "C:\ProgramData\hp\HP WebInspect\Policies\<custom_policy>.policy"` For more information on wi.exe options, see "Using wi.exe" on page 284.
3.	To view the results, open the scan in OpenText DAST. The scan name will be `API Assessment <API_Type> <Service_URL>`. **Note:** You cannot view the scan in OpenText DAST until the scan has completed.

Important considerations about definition files

Consider the following facts when configuring your scan settings file or constructing your CLI command:

- OpenText DAST attempts to generate the definition from the URL provided in the CLI command. It assumes that the API endpoint is the same URL, but without the file name. If your service is at the same location as your definition file, which is generally the case for GraphQL, then providing a URL will work. However, the definition may be in a different location for SOAP and gRPC.
- The GraphQL API must have introspection enabled to download the schema contents for the scan. If you do not want to enable introspection, you can perform a full schema query (an introspection query). You can then place the response into the APIDefinition setting in the JSON file.

Recommendations

Follow these recommendations when conducting an API scan using wi.exe:

- An API scan uses the API discovery engine regardless of the policy used. However, if the API discovery check is not enabled in the policy, then it will not appear in the findings. For this reason, OpenText recommends that you use a policy that has the API discovery check enabled.
- OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Understanding the API scan configuration file

The following table describes the parameters available for use in the JSON configuration file.

Important! You must escape all double quotation marks that are inside double quotation marks in the JSON file. Use one backslash (\) in front of each quotation mark to escape. For example:

```
"Setting": "Value \"Value Text Inside Quotes\""
```

Parameter	Description
`APIDefinition`	Points to the service definition location, which is a specific URL. Each API service uses a specific type of file, as follows: • SOAP uses a Web Service Definition Language (WSDL) file.

Parameter	Description
	• gRPC uses a `.proto` file . • GraphQL uses an introspection query or endpoint, such as `introspection-query.graphql`. The `APIDefinition` does not need to be a URL. It can be a URL or the contents of the API definition. For example, you can put a file path to the definition file on your local machine. If definition location points to an HTTP URL or a directory path, OpenText DAST downloads the content and replaces the URL or path with the content. Thus, the whole definition file is stored inside the settings.
`Type`	Indicates the type of API service being scanned. Possible values are: • `GraphQL` • `gRPC` • `SOAP`
`Schemes`	Indicates the protocol used by the service, either `http` or `https` or both. **Important!** Schemes must be defined as a JSON array regardless of whether one or many values are used. The following are examples of arrays: `[ "http" ], [ "http", "https" ]`
`Host`	Indicates the host name or URL where the service is running. **Tip:** This is most likely the same as the API definition root URL.
`APIVersion`	Primarily used for SOAP, allows filtering of operations by a specific version. Possible values are: • `Legacy` – filters against the lowest supported version. • `Mixed` – uses a combination of `Legacy` and `Newest`, depending on what is available. • `Newest` – the default setting, filters against the latest version.

Parameter	Description
`ServicePath`	Specifies the directory path to the service.
`AuthProviders`	Optionally, identifies the authentication type, such as a transport bearer token. For more information about `AuthProviders` parameter, see "Understanding API AuthProviders configuration" on page 163.
`Proxy`	Optionally, specifies proxy settings. `Proxy` requires the following parameters: • `Host` – Indicates the host name or URL where the proxy is running • `Port` – Indicates the port number used by the proxy server • `UserName` – Optionally, identifies the user account for accessing the proxy server • `Password` – Optionally, specifies the password for the user profile **Important!** Currently, only Basic authentication is supported.
`preferredContentType`	Optionally, sets the preferred content type of the request payload. If `preferredContentType` is in the list of supported content types for an operation, the generated request payload will be of that type. Otherwise, the first content type listed in an operation will be used.
`excludeOperations`	Optionally, defines a deny list of operation IDs that should be excluded from the output, expressed as an array of operation IDs. **Example:** `[ 'operation1', 'operation2', 'operationN' ]`
`includeOperations`	Optionally, defines an allow list of operation IDs that should be included in the output, expressed as an array of operation IDs . **Example:**

Parameter	Description
	`[ 'operation1', 'operation2', 'operationN' ]`
`parameterRules`	Optionally, defines specific values for a parameter when the default value is not appropriate or when the parameter is not defined in the API definition. **Example:** A parameter, such as an authorization header which is not defined in the API definition, needs to be injected into every request. The property is expressed as an array of 'parameterRule' objects. The 'parameterRule' objects are described in "Understanding parameter rule objects" below.

For sample JSON configuration files, see "API scan configuration file samples" on page 170.

Understanding parameter rule objects

The 'parameterRule' objects are described in the following table.

Object	Required / Optional	Description
`name`	Required	Specifies the parameter name to match. To override a property when you have a name conflict, specify the type of object from the API definition in front of the parameter name, separated by a slash in the format '*<type_of_object>*/*<parameter_name>*'. For example, if you have a parameter named “name” and a nested parameter also named “name”, you must specify the type of object for the nested parameter as shown below.

Object	Required / Optional	Description
		<pre>{ name : 'name', value : 'Romeo', location : 'body', type : 'string', includeOperations : ['addPet'] }, { name : 'tag/name', value : 'Juliet', location : 'body', type : 'string', includeOperations : ['addPet'] },</pre>
`value`	Required	Specifies the parameter value to substitute or inject.
`location`	Optional	Identifies the parameter location to match. Options are: • `'body'` • `'header'` • `'path'` • `'query'` • `'any'` The default is 'any' and matches all locations .
`type`	Optional	Identifies the parameter type to match. Options are: • `'number'` • `'boolean'` • `'string'` • `'file'` (See filename below.) • `'date'` • `'any'` The default is 'any' and matches all types.
`filename`	Optional	Replaces the filename attribute of a matching

Object	Required / Optional	Description
		multipart or form file entry. Valid only if `type` is `'file'`.
`inject`	Optional	Replaces parameter values. Options are: • `true` - injects the parameter in the specified location regardless of whether a matching name or type is found. • `false` - replaces only parameter values that match the specified name, location, and type. The default is false.
`base64Decode`	Optional	Specifies whether `'value'` is base64 encoded binary data. Options are: • `true` - `'value'` is assumed to be base64 encoded binary data and will be decoded into a byte array when inserted into a generated HTTP request. • `false` - `'value'` is not base64 encoded binary data. The default is false.
`includeOperations`	Optional	Applies this parameter rule to the operation IDs in the list, expressed an array of operation IDs. **Example:** `[ 'operation1', 'operation2', 'operationN' ]`
`excludeOperations`	Optional	Does not apply this parameter rule to the operation IDs in the list, expressed as an array of operation IDs. **Example:** `[ 'operation1', 'operation2', 'operationN' ]`

Understanding API AuthProviders configuration

You can configure the following categories of AuthProviders in your configuration file:

- Client certificate
- Message-based
- Transport (Authorization, Basic, Bearer, Digest, and NTLM)
- Transport custom header

You can combine AuthProvider types as needed to access the network and the API definition file or endpoint. With the exception of transport custom header, however, you can only use one AuthProvider from each category.

Client certificate

The following table describes the client certificate category of AuthProviders that you can configure in the scan configuration file.

Type	Description
`TRANSPORT_CERTIFICATE`	Uses certificate configuration for authentication. `TRANSPORT_CERTIFICATE` requires the following parameter: • `ClientCertificate` – Specifies the certificate information and includes the following parameters: • `Data` – Provides the base-64 encoded `*.pfx` certificate that is password protected. **Tip:** You can use the following openssl command to obtain a text version of the certificate to use as the value for `Data`: `openssl base64 -A -in d:\dump\cert.pfx -out d:\dump\cert.pfx.base64` • `Password` – Specifies the password for the certificate. The following sample shows the syntax for these parameters. `{ "Type": "TRANSPORT_CERTIFICATE", "ClientCertificate": { "Data": "<64-base_encoded_certificate>", "Password": "<Password>" }, }`

Message based

The following table describes the message-based category of AuthProviders that you can configure in the scan configuration file.

Type	Description
`MESSAGE_CERTIFICATE`	Uses certificate configuration for authentication. `MESSAGE_CERTIFICATE` requires the following parameter: • `ClientCertificate` – Specifies the certificate information for the client and includes the following parameters: • `Data` – Provides the base-64 encoded `*.pfx` certificate that is password protected. **Tip:** You can use the following openssl command to obtain a text version of the certificate to use as the value for `Data`: `openssl base64 -A -in d:\dump\cert.pfx -out d:\dump\cert.pfx.base64` • `Password` – Specifies the password for the certificate. • `ServerCertificate` – Specifies the certificate information for the server and includes the following parameter: • `Data` – Provides the base-64 encoded `*.pfx` certificate that is password protected. The following sample shows the syntax for these parameters. `{ "Type": "MESSAGE_CERTIFICATE", "ClientCertificate": { "Data": "<64-base_encoded_certificate>" "Password": "<Password>" }, "ServerCertificate": { "Data": "<64-base_encoded_certificate>" } }`
`MESSAGE_USERNAMETOKEN`	Uses a user name and token to authenticate to the SOAP service. `MESSAGE_USERNAMETOKEN` requires the following parameters: • `Username` – Identifies the user name whose credentials are used to

Type	Description
	access the SOAP service. • `Password` – Specifies the password for the user name. • `UsernameToken` – Provides message-level authentication for SOAP and includes the following parameters: • `Type` – Specifies the type of token. Options are `TEXT` and `HASH`. • `TimeStamp` – Optionally, indicates when the `usernameToken` was created and when it expires. `TimeStamp` accepts `Created`, `Full`, or `None`. • `IncludeNonce` – Indicates whether nonce is enabled. Nonce is required for `HASH` tokens because it helps the server to recalculate the hash and compare it to the data the client sent. Options are `true` or `false`. • `WSAddressing` - Optionally, identifies the Web Services Addressing (WS-Addressing) schema version used by the SOAP service. Options are `NONE`, `WSA_0408`, and `WSA_0508`. • `To` - Optionally, identifies the URL for the Web service host. **Note:** SOAP services may be exposed by way of a load balancer or reverse proxy. This configuration may prevent the sensor from getting the correct information for the internal Web service host name. The `To` URL override provides the correct address into WS Addressing. The following sample shows the syntax for these parameters. `{ "Type": "MESSAGE_USERNAMETOKEN", "Username": "<username>", "Password": "<password>", "UsernameToken": { "Type": "TEXT", "TimeStamp": "Created", "IncludeNonce": true }, "WSAddressing": { "Version": "WSA_0408", "To": "http://webservice/wcf/service.svc/CustomSoapEndpoint"`

Type	Description
	<pre>} }</pre>

Transport

The following table describes the transport category of AuthProviders that you can configure in the scan configuration file.

Type	Description
`TRANSPORT_AUTHORIZATION`	Uses an authorization token. `TRANSPORT_AUTHORIZATION` includes the following parameters: • `Name` – Indicates the token name. • `Value` – Provides the token value. The following sample shows the syntax for these parameters. <pre>{ "Type": "TRANSPORT_AUTHORIZATION", "Name": "<*token_name*>", "Value": "<*token_value*>", }</pre>`TRANSPORT_AUTHORIZATION` can also use the `Fetch` parameter. For more information, see "Fetching a token value" on page 169.
`TRANSPORT_BASIC`	Uses Basic configuration for authentication. `TRANSPORT_BASIC` requires the following parameters: • `Username` – Identifies the user name whose credentials are used to access the API service. • `Password` – Specifies the password for the user name. The following sample shows the syntax for these parameters. <pre>{ "Type": "TRANSPORT_BASIC", "Username": "<*UserName*>",</pre>

Type	Description
	<pre>"Password": "<Password>", }</pre>
TRANSPORT_BEARER	Uses a bearer token configuration for authentication. TRANSPORT_BEARER requires the following parameter: • Value – Provides the JSON token, generally from a response to a login form. • Header – Optionally, identifies a custom header name. The following sample shows the syntax for this parameter. <pre>{ "Type": "TRANSPORT_BEARER", "Value":"<token>" }</pre> TRANSPORT_BEARER can also use the Fetch parameter. For more information, see "Fetching a token value" on page 169.
TRANSPORT_DIGEST	Uses Digest authentication. TRANSPORT_DIGEST requires the following parameters: • Username – Specifies the host name and username combination whose credentials are used to access the API service. • Password – Specifies the password for the user name. The following sample shows the syntax for these parameters. <pre>{ "Type": "TRANSPORT_DIGEST", "Username": "<host_name>\\<username>", "Password": "<password>", }</pre>
TRANSPORT_NTLM	Uses NTLM configuration for authentication. TRANSPORT_NTLM requires the following parameters: • Username – Specifies the host name and username

Type	Description
	combination whose credentials are used to access the API service. • `Password` – Specifies the password for the user name. The following sample shows the syntax for these parameters. `{ "Type": "TRANSPORT_NTLM", "Username": "<host_name>\\<username>", "Password": "<password>", }`

Transport custom

The following table describes the transport custom category of AuthProviders that you can configure in the scan configuration file.

Important! OpenText recommends that you do not configure more than one custom header using the same HTTP header name.

Type	Description
`TRANSPORT_CUSTOM_HEADER`	Using this type, you can configure multiple custom headers. `TRANSPORT_CUSTOM_HEADER` includes the following parameters: • `Header` – Identifies the HTTP header name. The header must be unique and cannot be `Authorization`. • `Name` - Optionally, specifies the header value prefix name. • `Value` – Provides the header value. The following sample shows the syntax for these parameters. `{ "Type": "TRANSPORT_CUSTOM_HEADER", "Header": "<header_name>", "Name": "<prefix_header_name>", "Value": "<header_value>" }`

Type	Description
	The following sample shows the syntax for configuring the "X-MyCustomAuth: CustomToken value" HTTP custom header with a custom authentication token. `{` `"Type": "TRANSPORT_CUSTOM_HEADER",` `"Header": "X-MyCustomAuth",` `"Name": "CustomToken",` `"Value": "<token_value>"` `}`

Fetching a token value

The `TRANSPORT_AUTHORIZATION` and `TRANSPORT_BEARER` AuthProviders can use the fetch parameter, which generates a response from a workflow macro and uses a regular expression to capture a token for use in applying state. `Fetch` accepts the following parameters:

- `Macro` – Imports a base-64 encoded text capture of the macro. The macro must include a login web form and a response with a JSON web token.

  Tip: You can open the macro file in a text editor, and then copy and paste the text into the `Macro` field.

- `Search` – Allows use of a custom regular expression to fetch the token value. If a match to the regular expression occurs in the response, then the value is fetched and used as a bearer token. If the regular expression contains parentheses, then the value inside the parentheses will be extracted and used as a bearer token. Only the first value inside parentheses will be used.

  Important! The `Search` parameter must follow the `Macro` parameter in your JSON file.

- `IsolatedState` – Optionally, determines how the fetch macro playback is applied. When set to `true`, each scan thread runs its own fetch macro playback and applies the bearer token value to the thread. When set to `false`, only one fetch macro playback runs for all scan threads and the single shared bearer token value is applied to all threads. The default setting is `false`.

The following sample shows the syntax for these parameters.

```
{
     "Type": "TRANSPORT_BEARER",
     "Fetch": {
          "Macro": "<base-64_encoded_text>",
```

```
          "Search": "<regular_expression>",
          "IsolatedState": "true"
        },
}
```

API scan configuration file samples

The following paragraphs provide JSON samples of API configuration files.

Sample GraphQL configuration file

The following sample shows a GraphQL configuration file without authentication.

```
{
    "APIDefinition": "http://<ip_address>:<port>/graphql/",
    "Schemes": [ "http" ],
    "Host": "<ip_address>:<port>",
    "ServicePath": "/graphql/",
    "Type": "GraphQL",
    "Proxy": {
        "Host": "<ip_address>",
        "Port": "<port>",
        "UserName": "<username>",
        "Password": "<password>"
    }
}
```

Sample gRPC configuration file

The following sample shows a gRPC configuration file using the `TRANSPORT_BEARER` AuthProvider type with an explicitly configured token value.

```
{
   "APIDefinition": "https://<host_name>:<port>/protos/client.proto",
   "Type": "gRPC",
   "Schemes": [ "https" ],
   "Host": "<host_name>:<port>",
   "ServicePath" : "/",
   "AuthProviders": [
      {
         "Type": "TRANSPORT_BEARER",
         "Value":"<token>"
      }
```

```
   ]
}
```

Sample SOAP configuration file

The following sample shows a SOAP configuration file using the `TRANSPORT_NTLM` and `MESSAGE_USERNAMETOKEN` AuthProvider types.

```
{
   "APIDefinition": "https://<host_
name>:<port>/wcf/service.svc?singleWsdl",
   "Type": "SOAP",
   "Schemes": [ "https" ],
   "Host": "<host_name>:<port>",
   "APIVersion": "Mixed",
   "AuthProviders": [
      {
         "Type": "TRANSPORT_NTLM",
         "Username": "<host_name>\\<username>",
         "Password": "<password>",
      },
      {
         "Type": "MESSAGE_USERNAMETOKEN",
         "Username": "<username>",
         "Password": "<password>",
         "UsernameToken": {
            "Type": "TEXT",
            "TimeStamp": "Created",
            "IncludeNonce": true
         },

      }
   ]
}
```

Running a Basic Scan (website scan)

The options displayed by default on this and subsequent windows are extracted from the OpenText DAST 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 full complement of OpenText DAST settings, any selections you make are also temporary. To change the

default settings, you must select **Default Scan Settings** from the **Edit** menu. For more information, see "Default scan settings" on page 358.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Configuring Basic Scan options

1. In the **Scan Name** box, enter a name or brief description of the scan.
2. Select one of the following scan modes:
   - **Crawl Only**: Completely map 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**: Map the site's hierarchical data structure and audit 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 359.
   - **Audit Only**: Apply the methodologies of the selected policy to determine vulnerability risks, but do not crawl the website. No links on the site are followed or assessed.
   - **Manual**: Enables you to navigate manually to whatever sections of your application you choose to visit, using TruClient with Firefox. OpenText DAST 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.

     Note: Manual mode is not available when scheduling a scan.
3. Select a rendering engine from the **Rendering Engine** drop-down list. The rendering engine you select determines which Web Macro Recorder is opened when recording a new macro or editing an existing macro while configuring a scan. Options are as follows:
   - **Event-based (preferred)** – Selecting this option designates the Event-based Web Macro Recorder, which uses TruClient with Firefox technology.
   - **Session-based** – Selecting this option designates the Session-based Web Macro Recorder, which uses Internet Explorer browser technology.

Note: You cannot configure the Rendering Engine for Manual mode. Manual mode uses the TruClient with Firefox technology.

4. Select one of the following scan types:
   - **Standard Scan**: Performs an automated analysis, starting from the target URL. This is the normal way to start a scan.
   - **Manual Scan**: (also known as Step Mode) Enables you to navigate manually to whatever sections of your application you choose to visit, using TruClient with Firefox. This choice appears only if you select the Manual Scan mode.
   - **List-Driven Scan**: Performs 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.
     - To import a list, click **Import**.
     - To build or edit a list using the Site List Editor, click **Manage**. For more information, see "Using the Site List Editor" on page 186.
   - **Workflow-Driven Scan**: 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. If you select multiple macros, they will all be included in the same scan. You can use `.webmacro` files, Burp Proxy captures, or `.har` files. For more information, see "Selecting a workflow macro " on page 236.

     Important! If you use a login macro in conjunction with a workflow macro or startup macro or both, all macros must be of the same type: all `.webmacro` files or all Burp Proxy captures or all `.har` files. You cannot use different types of macros in the same scan.

5. Continue according to the following table.

If you selected...	Then follow these instructions...
Standard Scan	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, OpenText DAST 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

If you selected...	Then follow these instructions...
	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). OpenText DAST supports both Internet Protocol version 4 (IPV4) and Internet Protocol version 6 (IPV6). IPV6 addresses must be enclosed in brackets. For more information, see "Internet Protocol version 6" on page 356. 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** - OpenText DAST 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/, OpenText DAST will assess only the "two" directory. ◦ **Directory and subdirectories** - OpenText DAST 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** - OpenText DAST will begin crawling and/or auditing at the URL you specify, but will not access any directory that is lower in the directory tree. For information about limitations to the Restrict to folder scan option, see "Restrict to folder limitations" on page 196.
Manual Scan	Enter a **Start URL** and, if desired, select **Restrict to folder**. See Standard Scan described previously. **Note:** You cannot configure the Rendering Engine for Manual mode. Manual mode uses the TruClient with Firefox technology.
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.

If you selected...	Then follow these instructions...
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.

6. Click **Next**.
   The Authentication and Connectivity page appears.

Configuring network authentication and connectivity

On the Authentication and Connectivity page, you can configure proxy, network authentication, and site authentication settings.

Configuring proxy settings

To configure access to the target website through a proxy server:

1. Select **Network Proxy**.
2. Choose an profile from the **Proxy Profile** list. Profiles are:
   - **Auto Detect**: Use the Web Proxy Autodiscovery Protocol (WPAD) to locate a proxy autoconfig file and use this to configure the browser's web proxy settings.
   - **Use System Proxy**: Import your proxy server information from the local machine.
   - **Use PAC File**: Load proxy settings from a Proxy Automatic Configuration (PAC) file. If you select this option, click **Edit** to enter the location (URL) of the PAC. For more information, see "Configuring the proxy profile" on page 186.
   - **Use Explicit Proxy Settings**: Specify proxy server settings. If you select this option, click **Edit** to enter proxy information. For more information, see "Configuring the proxy profile" on page 186.
   - **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 Windows setting "Use a proxy server for your LAN" is not selected, then a proxy server will not be used.

Configuring network authentication

To configure network authentication to the target website:

1. Select **Network Authentication**.
2. Select an authentication method and enter your network credentials. The authentication methods are:
   - **ADFS CBT**
   - **Automatic**
   - **Basic**
   - **Digest**
   - **Kerberos**
   - **Negotiate**
   - **NT LAN Manager (NTLM)**
   - **OAuth 2.0 Bearer**
3. Do one of the following:
   - For all authentication methods except OAuth 2.0 Bearer, type a user ID in the **User Name** box and the user's password in the **Password** box.
   - For the OAuth 2.0 Bearer method, click **Configure** and continue with "Configuring OAuth 2.0 bearer credentials" on page 400.

Using client certificates

To use a client certificate for a website, click **Settings > Authentication** and continue as follows:

1. In the Client Certificates area, select the **Enable** check box.
2. Click **Select**.

   The Client Certificates window opens.
3. Do one of the following:
   - To use a certificate that is local to the computer and is global to all users on the computer, select **Local Machine**.
   - To use a certificate that is local to a user account on the computer, select **Current User**.

     Note: Certificates used by a common access card (CAC) reader are user certificates and are stored under Current User.

4. Do one of the following:
   - To select a certificate from the "Personal" ("My") certificate store, select **My** from the drop-down list.
   - To select a trusted root certificate, select **Root** from the drop-down list.
5. Does the website use a CAC reader or a certificate that is password protected?
   - If *yes*, do the following:
     i. Select a certificate that is prefixed with “(Protected)” from the **Certificate** list.
     Information about the selected certificate and a Password/PIN field appear in the Certificate Information area.
     ii. If a password or PIN is required, type it in the **Password/PIN** field.

     Note: If a password or PIN is required and you do not enter it at this point, you must enter the password or PIN in the Windows Security window each time it prompts you during the scan.

     Important! By default, OpenText DAST uses OpenSSL. If you are using a specific SSL/TLS protocol rather than OpenSSL, the Profiler portion of scan configuration may not work with certificates that are protected with a password.

     iii. Click **Test**.
     If you entered the correct password or PIN, a Success message appears.
   - If *no*, select a certificate from the **Certificate** list.
     Information about the selected certificate appears below the Certificate list.
6. Click **OK**.

Updating certificates in composite scan settings

The composite scan settings include a BIN file that contains encrypted certificate data. If you need to replace or update the client certificate in your composite scan settings, you can place the updated PFX or P12 file inside the certificates directory in the composite settings ZIP file. When OpenText DAST opens the settings, it will check for PFX and P12 files first. If none are present, then the BIN file will be decrypted and used. For more information about composite settings, see "Application settings: General" on page 431.

To replace or update the client certificate:

1. Locate the encrypted BIN file in the certificates directory in your composite scan settings ZIP. The file name is a GUID, similar to the following:
   `<your-scansettings.zip>\certificates\0b627638-efda-4d01-a83e-80ee3a79b4cf.bin`

Note: The default settings file location in Windows is `C:\ProgramData\HP\HP WebInspect\Settings\`.

2. Place the updated PFX or P12 file into the same directory.
3. Rename the PFX or P12 file the same name as the BIN file. Using the previous example, the file name will be as follows:

   `0b627638-efda-4d01-a83e-80ee3a79b4cf.pfx`

   – OR –

   `0b627638-efda-4d01-a83e-80ee3a79b4cf.p12`

   Important! Be sure to retain the original file extension.

4. Optionally, if you want to retain an encrypted certificate in the settings, save the settings again and the BIN file will reflect the updated PFX or P12 certificate. The PFX or P12 certificates will be removed from the ZIP.

Tip: PFX and P12 certificates frequently require a password. Use one of the following options to provide the password for the settings:

- Set an empty password when you create the PFX or P12 certificate and place it in the settings ZIP file.
- Keep the password for the PFX or P12 certificate and edit the `settings.json` file to place the password as the value for the CertificatePin as follows:

  `"CertificatePin": "<password>"`

Configuring site authentication

To configure site authentication:

1. Select **Site Authentication** to use a recorded macro containing one or more usernames and passwords that enables you to log in to the target site. The macro must also contain a "logout condition," which indicates when an inadvertent logout has occurred so OpenText DAST can rerun this macro to log in again.

   If the macro uses parameters for which values are masked in the Web Macro Recorder, then these values are also masked when configuring a Basic Scan in OpenText DAST.

   If **Enable macro validation** is selected in Scan Settings: Authentication for scans that use a login macro, OpenText DAST tests the login macro at the start of the scan to ensure that the log in is successful. If the macro is invalid and fails to log in to the application, the scan stops and an error message is written in the scan log file. For more information and troubleshooting tips, see "Testing login macros" on page 494.

   Note: Macro testing is not supported for macros containing two-factor authentication.

Important! If you use a macro that includes Two-factor Authentication, then you must configure the Two-factor Authentication Application settings before starting the scan. For more information, see "Application settings: Two-Factor Authentication" on page 442.

Continue according to the following table.

To...	Then...
Use a pre-recorded Web Macro Recorder macro	Click the ellipsis button (...) to select a macro. If, after selecting the macro, you want to modify it using the Web Macro Recorder, click **Edit**. **Tip:** To erase the macro name, clear the **Site Authentication** check box.
Create a new macro	Click **Record**. The Web Macro Recorder opens. **Note:** For more information about using the Web Macro Recorder, see the Web Macro Recorder Help.
Automatically create a login macro **Note:** You cannot automatically create login macros for privilege-escalation and multi-user login scans.	a. Select **Auto-gen Login Macro**. b. Type a username in the **Username** field. c. Type a password in the **Password** field. Optionally, click **Test** to locate the login form, generate the macro, and run macro validation tests before advancing to the next stage in the Scan wizard. If you need to cancel the validation test prior to completion, click **Cancel**. If the macro is invalid and fails to log in to the application, an error message appears. For more information and troubleshooting tips, see "Testing login macros" on page 494.

2. Click **Next**.
   The Crawl Coverage and Thoroughness page appears.

Configuring crawl coverage and thoroughness

To configure a balance between efficiency 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. Fortify may develop other settings overlays and make them available through Smart Update.

   For more information about scanning a WebSphere portal, see "WebSphere Portal FAQ " on page 281.
2. Use the **CrawlCoverage** 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 enables you to select one of four crawl positions. Each position represents a specific collection of settings, as represented by the following labels:

   Thorough

   A Thorough crawl is an automated crawl that uses the following settings:
   - Redundant Page Detection: **OFF**
   - Maximum Single URL Hits: **10**
   - Maximum Web Form Submissions: **7**
   - Maximum Script Events Per Page: **2000**
   - Number of Dynamic Forms Allowed Per Session: **Unlimited**
   - Include Parameters In Hit Count: **True**

   Default

   A Default crawl is an automated crawl that uses the following (default scan) settings:
   - Redundant Page Detection: **OFF**
   - Maximum Single URL Hits: **5**
   - Maximum Web Form Submissions: **3**
   - Maximum Script Events Per Page: **1000**
   - Number of Dynamic Forms Allowed Per Session: **Unlimited**
   - Include Parameters In Hit Count: **True**

   Moderate

   A Normal crawl is an automated crawl that uses the following settings:
   - Redundant Page Detection: **OFF**
   - Maximum Single URL Hits: **5**
   - Maximum Web Form Submissions: **2**
   - Maximum Script Events Per Page: **300**

- Number of Dynamic Forms Allowed Per Session: **1**
- Include Parameters In Hit Count: **False**

Quick

A Quick crawl uses the following settings

- Redundant Page Detection: **ON**
- Maximum Single URL Hits: **3**
- Maximum Web Form Submissions: **1**
- 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 dialog box) 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. Click **Next**.
   The Audit Coverage and Thoroughness page appears.

Configuring audit coverage and thoroughness

You can select a different policy than the default selection or you can configure multiple policies for better coverage or for additional focus on a specific type of vulnerability. For example, if you want to run a scan using the Standard policy, but want additional focus on SQL Injection, you can select the Standard policy and the SQL Injection policy for the scan. The sensor aggregates all selected policies during the scan.

To select a different policy:

1. In the **Audit Depth (Policy)** list, slide the toggle for the selected policy to the disabled position.
2. Slide the toggle for the desired policy to the enabled position.
   The selected policy appears in the ENABLED SCAN POLICIES list. For descriptions of policies, see "OpenText DAST policies" on page 458.
3. Click **Next**.
   The Detailed Scan Configuration page appears. If the Run Profiler Automatically option is selected, a "Profiling Site..." message appears.

To select additional policies:

1. In the **Audit Depth (Policy)** list, slide the toggle for the desired policies to the enabled position.

The selected policies appear in the ENABLED SCAN POLICIES list. For descriptions of policies, see "OpenText DAST policies" on page 458.

2. Click **Next**.

   The Detailed Scan Configuration page appears. If the Run Profiler Automatically option is selected, a "Profiling Site..." message appears.

Using the Profiler

On the Detailed Scan Configuration page, OpenText DAST conducts a preliminary examination of the target website 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 OpenText DAST should not conduct "file-not-found" detection. This process is useful for websites 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 OpenText DAST 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**. For more information, see "Server Profiler" on page 238.

Results appear in the Settings section.

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."

Choosing Profiler suggested settings

You can choose to accept or reject the suggested settings in the Settings area of the Detailed Scan Configuration page.

Several options may be presented even if you do not run the Profiler. They include:

- Auto fill Web forms (see "Auto fill web forms" on the next page)
- Add allowed hosts (see "Add allowed hosts" on the next page)
- Reuse identified suppressed findings (see "Reuse identified Suppressed Findings" on the next page)

- Apply sample macro (see "Sample macro" on the next page)
- Traffic analysis (see "Traffic analysis" on the next page)

To accept or reject Profiler suggestions:

1. Accept or reject the suggestions as follows:
   - To accept a setting, select the associated check box.
   - To reject a setting, clear the associated check box.
2. If necessary, provide the requested information.
3. Click **Next**.
   The Congratulations window appears.

Auto fill web forms

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

- Click the ellipsis button ... 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. For more information, see "Scan settings: Allowed Hosts" on page 375.

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**.

For more information about adding or editing Allowed Hosts, see "Specifying allowed hosts" on page 187.

Reuse identified Suppressed Findings

You can import vulnerabilities that were changed to false positive or ignored in previous scans. If those false positive or ignored items match vulnerabilities detected in the current scan, the vulnerabilities will be changed to false positive or ignored. You can

import suppressed findings from existing scans or suppressed findings files. For more information, see "Suppressed findings" on page 67.

To reuse identified suppressed findings:

1. Select **Import Suppressed Findings**.
2. Continue according to the following table.

To use...	Then...
Existing scans	a. Click **select scans**. The Select a Scan to Import Suppressed Findings dialog opens. b. Select one or more scans containing suppressed findings from the same site you are now scanning. c. Click **OK**.
Suppressed findings files	a. Click **select file**. A standard Windows file selection dialog box opens. b. Select the file to import, and then click **Open**. c. Optionally, repeat Steps a and b to select additional files.

Note: You cannot import suppressed findings when scheduling a scan.

Sample macro

OpenText DAST's example banking application, zero.webappsecurity.com, uses a Web form login. If you scan this site, select **Apply sample macro** to run the sample 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 OpenText DAST and the responses returned by the target server.

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

Congratulations

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

Saving settings

You can save the settings you configured for this scan, which would allow you to reuse the settings for a future scan.

Generating reports

If you are scheduling a scan, you can instruct OpenText DAST 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_2022 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 then 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**.

Using the Site List Editor

When performing a List-Driven Scan using the Basic Scan Wizard, you can build or edit the list of URLs using the Site List Editor.

To access the Site List Editor:

- Click **Manage** under the List-Driven Scan option in the Basic Scan Wizard.

To add individual URLs manually:

1. Click **Add**.
2. Enter a URL that you want to include in the scan. If you do not specify the protocol, the editor will add "http://" to the beginning of the URL.
3. Repeat as necessary.

To add URLs specified in a text file or XML file:

1. Click **Import**.
2. Using the standard file-selection window, locate the file and click **Open**.
3. Repeat as necessary.

> **Note:** The editor does not check for duplicates. If you import two lists and both lists contain the same URL, that URL will be listed twice.
>
> Also, each URL must include the protocol (for example, http:// or https://). Unlike manual entry, the editor will not automatically add a protocol to the beginning of an imported URL.

To edit an entry:

- Click a URL.

To delete an entry:

- Select a URL and click **Delete**.

See also

"Running a Basic Scan (website scan)" on page 171

Configuring the proxy profile

When performing a Basic Scan and using proxy settings from a Proxy Automatic Configuration (PAC) file or specifying Explicit Proxy Settings, you can configure the proxy options in the Proxy Profile window.

To access the Proxy Profile window:

- Click **Edit** under Network Proxy in the Basic Scan Wizard.

Configure proxy using a PAC file

Load proxy settings from a Proxy Automatic Configuration (PAC) file. Specify the file location 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**
   - **Standard**
3. If authentication is required, select a type from the **Authentication** list:
   - **Automatic**
   - **Basic**
   - **Digest**
   - **Kerberos**
   - **Negotiate**
   - **NT LAN Manager (NTLM)**

   Important! Socks4 proxy servers do not support authentication. When using a Socks proxy server that requires authentication, you must use a Socks5 proxy.
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.

See also

"Running a Basic Scan (website scan)" on page 171

Specifying allowed hosts

Specify an allowed host to add domains to be crawled. 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 or 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 OpenText DAST 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, OpenText DAST 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.

Specifying allowed hosts

To specify (add) allowed hosts:

1. On the Detailed Scan Configuration page of the Basic Scan Wizard, click **Add**.
2. On the Specify Allowed Host dialog box, enter a URL (or a regular expression representing a URL).

   Note: When specifying the URL, do not include the protocol designator (such as http:// or https://).

3. If you entered a regular expression for the allowed host, select **Use Regular Expression**.

   For assistance creating a regular expression, click  (to the right of the **Allowed Host** box).
4. Click **OK**.

Editing allowed hosts

To edit allowed hosts:

1. On the Detailed Scan Configuration page of the Basic Scan Wizard, select a host and then click **Edit**.
2. On the Edit Allowed Host dialog box, edit the URL (or the regular expression representing the URL).

Note: When editing the URL, do not include the protocol designator (such as http:// or https://).

3. Click **OK**.

See Also

"Running a Basic Scan (website scan)" on page 171

Multi-user login scans

Applications that allow only a single active login session per user prevent multi-threaded scanning. With multiple logins, the threads invalidate each other's state, resulting in slow scan times.

A solution to this problem is to convert the recorded credentials in a login macro to parameters and use multiple login accounts with the same application privileges. You can use the Multi-user Login option in the Scan Settings: Authentication window to parameterize the username and password in a login macro, and define multiple username and password pairs to use in a scan. You can also parameterize the phone number, email, and email password if two-factor authentication is required.

This approach allows the scan to run across multiple threads. Each thread has a different login session, resulting in faster scan times.

Before you begin

You must use a parameterized login macro to configure a multi-user login scan. For more information, see the "Working with Parameters" topic in the Web Macro Recorder chapters of the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Known limitations

The following known limitations apply to the multi-user login feature:

- When using this feature, OpenText DAST does not detect several login-related Securebase checks.
- This feature currently supports only shared requestor threads. Using default scan settings with separate crawl and audit threads is not supported. For more information, see "Scan settings: Requestor" on page 369.
- The scan does not distribute the work equally among the multiple users logged in. For example, one configured user might use up to 75% of the scan activities while all other users are allocated to the remaining 25% of scan activities.

Process overview

To configure a multi-user login scan, use the process described in the following table.

Stage	Description
1.	Set the shared requestor to the desired number of users. For more information, see "Scan settings: Requestor" on page 369. **Important!** The number of shared requestor threads should not be more than the number of configured users. Requestor threads without valid users will cause the scan to run longer. Remember to count the original username and password in the parameterized macro as the first user when you configure multiple users.
2.	Ensure that you have a login macro with parameterized username and password. Optionally, parameterize the phone number, email, and email password if two-factor authentication is required. For more information, see the "Working with Parameters" topic in the Web Macro Recorder chapters of the *OpenText™ Dynamic Application Security Testing Tools Guide*.
3.	In the Basic Scan wizard or Guided Scan wizard, enable the multi-user checkbox as described in "Configuring a multi-user login scan" below.
4.	Add credentials for multiple users as described in "Adding credentials" on the next page.
5.	Continue through the scan wizard as normal and conduct the scan.

Configuring a multi-user login scan

To configure a multi-user login scan:

1. Do one of the following:
   - From the Basic Scan wizard, click **Edit > Current Scan Settings**. Then, select **Scan Settings > Authentication**.
   - From the Guided Scan wizard, click **Advanced** in the ribbon, and then select **Scan Settings > Authentication**.
2. Select the **Use a login macro for forms authentication** checkbox.

   Important! You must select this checkbox to enable the multi-user login option.

3. Do one of the following:
   - To record a new macro, click **Record** and record a login macro as usual.

     Note: The Record button is not available for Guided Scan, because Guided Scan includes a separate stage for recording a login macro. After recording the macro, you must parameterize the credentials.

   - To use an existing macro, click **...** and select a saved macro that already has parameterized credentials.
4. Select the **Multi-user Login** checkbox.

   Note: If you clear the Multi-user Login checkbox prior to running the scan, the additional credentials will not be used during the scan. OpenText DAST will use only the original credentials recorded in the login macro.
5. Continue as follows:
   - To add a user's credentials, go to "Adding credentials" below.
   - To edit a user's credentials, go to "Editing credentials" on the next page.
   - To delete a user's credentials, go to "Deleting credentials" on the next page.
6. After configuring the user's credentials, continue through the scan wizard as normal and conduct the scan.

Adding credentials

To add credentials:

1. Under **Multi-user Login**, click **Add**.
   The Multi-user Credential Input dialog box appears.
2. In the **Username** box, type a username
3. In the **Password** box, type the corresponding password.
4. Optionally, if two-factor authentication is required, then continue according to the following table.

For this credential box...	Enter this...
Phone Number	Corresponding phone number for the username (to receive SMS responses)
Email	Corresponding email address for the username (to receive email responses)
Email Password	Password for the email address (to receive email responses)

5. Click **OK**.
6. Repeat Steps 1-5 for each user login to add.

Important! The number of shared requestor threads should not be more than the number of configured users. Requestor threads without valid users will cause the scan to run longer. Remember to count the original username and password in the parameterized macro as the first user when you configure multiple users. For more information, see "Scan settings: Requestor" on page 369.

Editing credentials

To edit credentials:

1. Under **Multi-user Login**, select an entry in the table and click **Edit**.
   The Multi-user Credential Input dialog box appears.
2. Edit the credentials as needed.
3. Click **OK**.

Deleting credentials

To delete credentials:

1. Under Multi-user Login, select an entry in the table to be removed.
2. Click **Delete**.

Using two-factor authentication

Two-factor authentication augments the standard password, which is defined as the "something you know" factor, with one of the following:

- Something you have, such as a one-time passcode (OTP) sent by SMS or email
- Something you are, such as your fingerprint, face, or retina

While this second factor of authentication improves security, it adds a layer of complexity when conducting an automated scan of web applications that implement it.

OpenText engineers have developed a method and process that enable OpenText DAST and the Event-based Web Macro Recorder to automate the "something you have" factor of two-factor authentication.

How scanning with two-factor authentication works

OpenText DAST includes a Node.js server that you configure for a control center to process the SMS and email responses coming from your application server. There is

also a mobile application that forwards SMS responses to the control center. The control center queues the responses and forwards them to the appropriate TruClient browser when needed for authentication.

Recommendation

We strongly recommend that you use test phones and test email addresses only. For privacy concerns, do not use personal phones and email addresses.

Known limitations

The following known limitations apply to the two-factor authentication feature:

- IMAP and POP3 servers are supported. However, only POP3 servers that support unique ID listing (UIDL) are supported.
- Currently, login macros with two-factor authentication using email support only the Basic authentication method for IMAP or POP3.
- Currently, only Android mobile phones are supported.
- The mobile phone requires a Wi-Fi connection in the same subnet where OpenText DAST is installed.

Facts about Gmail accounts

Be aware of the following facts related to Gmail accounts:

- Gmail account settings include normal mode and recent mode. If you use a Gmail account and experience issues with new incoming emails, using recent mode might resolve this issue. To enable recent mode, configure the account name in your POP3 account settings using the following format:

  `recent:<email_address@gmail.com>`
- For security, Google uses "Sign in with Google" to connect Gmail to a user's Google account and does not accept user-created passwords. When using a Gmail account, you must create and use a Google app password. For more information, refer to Google account documentation for creating and using app passwords.

Understanding the process

The following table describes the process for conducting a scan using two-factor authentication.

Stage	Description
1.	In the OpenText DAST application settings for two-factor authentication, do the following:

Stage	Description
	• Configure the two-factor authentication control center • Configure the mobile application (if SMS responses are used) For more information, see "Application settings: Two-Factor Authentication" on page 442.
2.	In the Event-based Web Macro Recorder, record a login macro and modify it as follows: 1. Add and configure a **Two-factor authentication** group step. **Note:** You must configure the group step for **SMS** or **email** responses. The group step includes a **Wait for 2FA** step that you must also configure. 2. Optionally, create username, password, phone number, email, and email password parameters. Using parameters for two-factor authentication enables you to conduct a multi-user login scan. 3. Configure the **Wait for 2FA** step. 4. Add a **Generic Object Action** step and configure it as a **Type** step. 5. Add a **Generic Object Action** step and configure it as a **Click** step. For more information, see the *OpenText™ Dynamic Application Security Testing Tools Guide*.
3.	In the Web Macro Recorder, replay the login macro.
4.	Optionally, if conducting a multi-user login scan, add credentials for username, password, phone number, email, and email password in the Scan Settings: Authentication window. For more information, see "Multi-user login scans" on page 189 and "Scan settings: Authentication" on page 393.
5.	In OpenText DAST, run a scan using the macro.

Interactive scans

Web applications using certain types of anti-scanning technology, such as CAPTCHA, require an interactive scan configuration in OpenText DAST. In an interactive scan, you are presented with a browser window asking for user input for authentication. You can configure an automated interactive scan that will pause only when an input field is

encountered. This pause affects only the Requestor thread that encounters the input field. The remaining threads are unaffected.

Interactive scan configuration works for CAPTCHA, RSA ID token fields, virtual PIN pads, virtual keyboards, and common access card (CAC) readers where the PIN or input is dynamic and changes.

Tip: For websites that use a CAC reader with a static PIN, you can configure the scan to use CAC certificates. See one of the following topics:

- "Scan settings: Authentication" on page 393
- "Running a Basic Scan (website scan)" on page 171
- "Using the Native Scan template" on page 123
- "Using the Mobile Scan template" on page 110
- "Using the Predefined template" on page 97

Note: Two-factor authentication does not require an interactive scan. You can configure fully-automated scans using two-factor authentication. For more information, see "Using two-factor authentication" on page 192.

Configuring an interactive scan

The following table describes the process for configuring an interactive scan.

Stage	Description
1.	Prepare the web forms input file as follows: 1. Record or enter the field name into the Web Form Editor tool. 2. Right-click the form name and select **Mark As Interactive**. 3. Save the web forms input file. For more information, see the Web Form Editor chapter in the *OpenText™ Dynamic Application Security Testing Tools Guide*.
2.	Are you using a client-side certificate that requires a dynamic PIN? • If *yes*, ensure that the client-side certificate is listed in the certificates list in your browser or manually import it. For example, in Microsoft Edge, access the **Manage certificates** dialog box in the **Privacy, search, and services** settings. This action temporarily loads the certificate into the Windows certificate store. **Note:** Plugging in the hardware token and entering the requested

Stage	Description
	PIN may do this automatically. • If *no*, skip to Stage 3.
3.	Configure the scan method for interactive scan mode as follows: 1. Open the Scan Settings: Method window. 2. In the **Auto fill web forms** field, specify the web forms input file you created in Stage 1. 3. Select the **Prompt for web form values during scan (interactive mode)** check box. 4. Select the **Only prompt for tagged inputs** check box. **Note:** If this final check box is not selected, you will be prompted for all inputs encountered on the site.
4.	Are you using a client-side certificate that requires a dynamic password or PIN? • If *yes*, configure authentication to use the client-side certificate: a. Open the Scan Settings: Authentication window. b. In the Client Certificates area, select the **Enable** check box and browse to select the user's certificate. OpenText DAST uses this certificate until it times out and fails to enter the requested password or PIN, or until the hardware token is removed and Windows drops the certificate from the store. • If *no*, skip to Stage 5.
5.	Save the scan settings and use them in an OpenText DAST scan. **Important!** You must watch for the pop-ups to enter the form value as needed.

Restrict to folder limitations

This topic describes limitations to the Restrict to folder scan option when JavaScript include files are encountered or when a login or workflow macro is used.

JavaScript include files

During a scan, the crawler and JavaScript engine might access external JavaScript include files. These files are not actively audited, so no attacks are sent over HTTP. However, passive inspection can reveal issues with JavaScript include files, and these files will be listed in the site tree.

Login macros

If you use a login macro, then sessions requested in the macro will be listed in the site tree. The sessions will be passively audited, meaning that no attacks will be sent, but vulnerabilities such as weak encryption, unencrypted login forms, and so on might be revealed.

Workflow macros

If you use a workflow macro in a Crawl and Audit scan or a Crawl Only scan, then the scan might violate the Restrict to folder option. The assumption is that you wish to visit the URLs included in the workflow macro.

Running a manual scan

A manual scan (also referred to as Step Mode) is a Basic Scan option that enables you to navigate manually to whatever sections of your application you choose to visit, using TruClient with Firefox. 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.

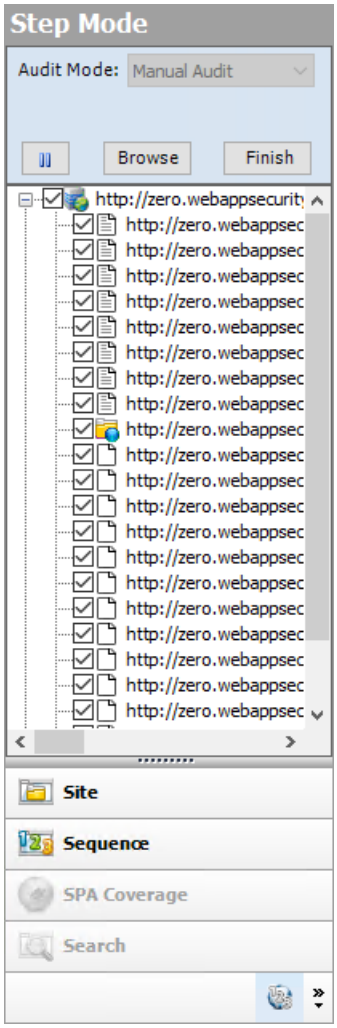
To conduct a manual scan:

1. On the OpenText DAST Start Page, select **Start A Basic Scan**.
2. Follow the instructions for configuring a Basic Scan as described in Basic Scan Wizard, selecting **Manual** as the scan method. For more information, see "Running a Basic Scan (website scan)" on page 171.
3. Click **Scan**.
4. When the browser opens, use it to navigate through the site, visiting the areas you want to record.

   Tip: If you want to visit certain areas of the application without recording the sessions, return to OpenText DAST and click the **Pause** button displayed in the Step Mode view of the Navigation pane. To resume recording sessions, click

the **Record** button . For more information, see "Navigation pane" on page 46.

5. When done, close the browser.
   OpenText DAST displays the Step Mode view in the Navigation pane.



6. Do one of the following:
   - To resume browsing the application, select a session and click **Browse**.
   - To import the sessions into the scan, click **Finish**. You can exclude an individual session from the import by clearing its associated check box.
7. To audit the recorded sessions, click Audit (on the toolbar).

About privilege escalation scans

Privilege escalation vulnerabilities result from programming errors or design flaws that grant an attacker elevated access to an application and its data. OpenText DAST can detect privilege escalation vulnerabilities by conducting either a low-privilege or

unauthenticated crawl followed by a high-privilege crawl and audit in the same scan. OpenText DAST includes a privilege escalation policy as well as privilege escalation checks that can be enabled in other policies, including custom policies. In Guided Scan, OpenText DAST automatically detects when you have selected a policy with privilege escalation checks enabled, and prompts you for the required login macro(s).

Two modes of privilege escalation scans

OpenText DAST can perform privilege escalation scans in two modes, determined by the number of login macros you use:

- Authenticated Mode – This mode uses two login macros: one for low-privilege access and one for high-privilege access. In this mode, a low-privilege crawl is followed by a high-privilege crawl and audit. You can perform this type of scan using Guided Scan. For more information, see "Running a Guided Scan" on page 96.
- Unauthenticated Mode – This mode uses only a high-privilege login macro. In this mode, the low-privilege crawl is actually an unauthenticated crawl. Any privilege escalation detected during this scan is moving from unauthenticated to high privilege. You can perform this type of scan using Guided Scan (and providing only a high-privilege login macro) or the Basic Scan wizard. For more information, see "Running a Basic Scan (website scan)" on page 171.

What to expect during the scan

When conducting a scan with privilege escalation checks enabled, OpenText DAST first performs a low-privilege crawl of the site. During this crawl, the Site view is not populated with the hierarchical structure of the website. Nor are vulnerabilities populated in the Summary pane. However, you can confirm that the scan is actively working by clicking the Scan Log tab in the Summary pane. You will see messages in the log indicating the "Scan Start" time and the "LowPrivilegeCrawlStart" time. When the low-privilege crawl of the site is complete, the high-privilege crawl and audit phase of the scan occurs. During this phase, the Site view will be populated and any vulnerabilities found will appear in the Summary pane. For more information, see "Summary pane" on page 87.

Regex patterns used to identify restricted pages

If your site includes restricted pages that are blocked using text such as “Forbidden,” “Restricted,” or “Access Denied,” the privilege escalation check includes a regex pattern that determines that these pages are forbidden for the current user. Therefore, these pages are not identified as being vulnerable for privilege escalation. However, if your site uses other privilege restriction text that does not match the built-in regex pattern, you must modify the regex to include your own text patterns. Otherwise, the privilege escalation check may generate false positives for those pages.

Modifying regex for privilege restriction patterns

To modify the regex:

1. Click **Edit > Default Scan Settings**.
   The Default Settings window appears.
2. Select **Attack Exclusions** in the Audit Settings group.
3. Click **Audit Inputs Editor...**.
   The Audit Inputs Editor appears.
4. Select **Check Inputs**.
5. Select check **11388 Privilege Escalation**.
   The Privilege Restriction Patterns appear in the right pane. By default, the pattern is as follows:

```
'forbidden|restricted|access\sdenied|(?:operation\snot\s
(?:allowed|permitted|authorized))|(?:you\s(?:do\snot|don't)\shave\s
(?:access|permission|authorization))|(?:you\s(?:are\snot|aren't)\s
(?:allowed|permitted|authorized))'
```

6. Using regex syntax, add any new forbidden action words that are used in your site.
7. Click **OK** to save the revised Check Inputs.
8. Click **OK** to close the Default Settings window.

Effect of crawler limiting settings on privilege escalation scans

OpenText DAST audits each parameter value during a scan. Therefore, a privilege escalation scan is sensitive to settings that limit the crawler, such as:

- Limit maximum single URL hits to
- Include parameters in hit count
- Limit maximum web form submission to
- Perform redundant page detection

For example, if you set "Limit maximum single URL hits to" 1 and the site contains links such as:

```
index.php?id=2
 index.php?id=1
 index.php?id=3
```

then during the high-privilege scan, OpenText DAST finds "`index.php?id=1`" and during the low-privilege scan, it finds "`index.php?id=3`". In this scenario, OpenText DAST will mark "`index.php?id=1`" with a privilege escalation vulnerability. This vulnerability will be a false positive.

For more information, see "Scan settings: General" on page 361.

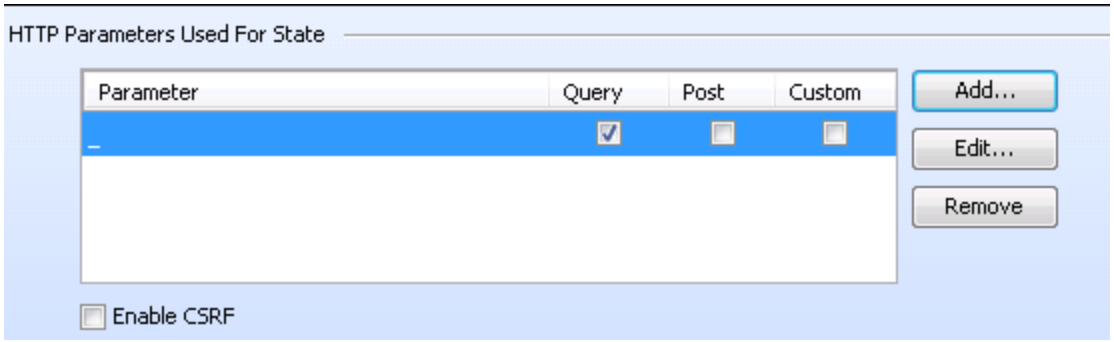
Effect of parameters with random numbers on privilege escalation scans

If the site contains parameters with random numbers, you can add the parameter to the list of HTTP Parameters Used For State to exclude such sessions from audit and reduce the number of false positives.

For example, for the following parameter:

```
index.php?_=1440601463586
 index.php?_=1440601465662
 index.php?_=1440601466365
```

you would add the parameter to the list of HTTP Parameters Used For State as shown below:



For more information, see "Scan settings: HTTP Parsing" on page 376.

See also

"Running a Basic Scan (website scan)" on page 171

"Using the Predefined template" on page 97

"Using the Mobile Scan template" on page 110

"Using the Native Scan template" on page 123

About single-page application scans

This topic describes single-page application (SPA) support for crawling and auditing the Document Object Model (DOM) of an application.

The challenge of single-page applications

Developers use JavaScript frameworks such as Angular, Ext JS, and Ember.js to build SPAs. These frameworks make it easier for developers to build applications, but more difficult for security testers to scan those applications for security vulnerabilities.

Traditional sites use simple back-end server rendering, which involves constructing the complete HTML web page on the server side. SPAs and other “Web 2.0” sites use front-end DOM rendering, or a mix of front-end and back-end DOM rendering. With SPAs, if the user selects a menu item, the entire page can be erased and recreated with new content. However, the event of selecting the menu item does not generate a request for a new page from the server. The content update occurs without reloading the page from the server.

With traditional vulnerability testing, the event that triggered the new content might destroy other events that were previously collected on the SPA for audit. Through its SPA support, OpenText DAST offers a solution to the challenge of vulnerability testing on SPAs.

Enabling SPA support

When you enable SPA support, the DOM script engine finds JavaScript includes, frame and iframe includes, CSS file includes, and AJAX calls during the crawl, and then audits all traffic generated by those events.

You can enable SPA support in the scan settings or in Guided Scan.

Caution! SPA support should be enabled for single-page applications only. Enabling SPA support to scan a non-SPA website will result in a slow scan.

See also

"Scan settings: JavaScript" on page 367

"Using the Predefined template" on page 97

"Using the Mobile Scan template" on page 110

"Using the Native Scan template" on page 123

Scan status

Unless otherwise specified, the scan status is read directly from the database. Scan statuses are described in the following table.

Tip: For most scan statuses, you can find the reason for the status in the scan log.

Status	Description
Completed	The scan has finished.
Incomplete	The user has paused the scan and closed it. The scan has not finished running.
Interrupted	There is an environmental issue, such as a connectivity issue with the application under test or with the database.
Locked	One instance of OpenText DAST is running a scan while connected to a SQL Server and a second instance of OpenText DAST connected to the same SQL Server database has tried to access the same scan. **Note:** Applies to remote SQL Server (full version) only.
Open	A user on the local machine has the scan open in OpenText DAST. The user may be the current user (in which case, the scan can be seen on the Scan tab) or it may be another user on the same machine (when using Terminal services, for example). The state stored in the scan database is ignored.
Paused	The user paused the scan.
Running	The scan is currently running on the local machine. **Note:** This status includes scheduled scans and those scans initiated through the command-line interface (CLI).

Updates to information in the scan manager

The scan manager is not intended to give real-time status information on any of the scans currently being displayed, with three notable exceptions:

- A new scan has been created or opened. In this case, the scan manager will list the new scan with a status of Open.
- A scan that was previously opened by the current user is closed. For example, a user opens/creates a scan, then closes it. The status in the scan manager for the scan is updated to reflect the status of the scan at the time it was closed (for example, Completed, Incomplete, etc.). All statistics will be refreshed for the single scan only.
- The duration field is not always accurate or available while a scan is open. Therefore, when a scan is in the Open, Running, or Locked state, the **Duration** column will show that the value is unavailable (instead of a number the user will see "-").

To see any other status changes or updated count information, the user MUST click the refresh button.

See also

"Scheduled scan status" on page 217

Opening a saved scan

Use one of the following procedures to open a saved file containing the results of a previous scan.

Using the Menu or Tool bar:

- Click **File > Open > Scan**.
- Click the drop-down arrow on the **Open** button and select **Scan**.

From the Start Page tab:

- Click **Start a Basic Scan**.
- On the Home pane, click an entry in the **Recently Opened Scans** list.
- On the Manage Scans pane, select a scan and click **Open** (or double-click the scan name).

OpenText DAST loads the scan data and displays it on a separate tab.

Comparing 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 the same site after 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 or SQL Server Standard or Enterprise Edition).

Selecting scans to compare

To compare two scans, do one of the following:

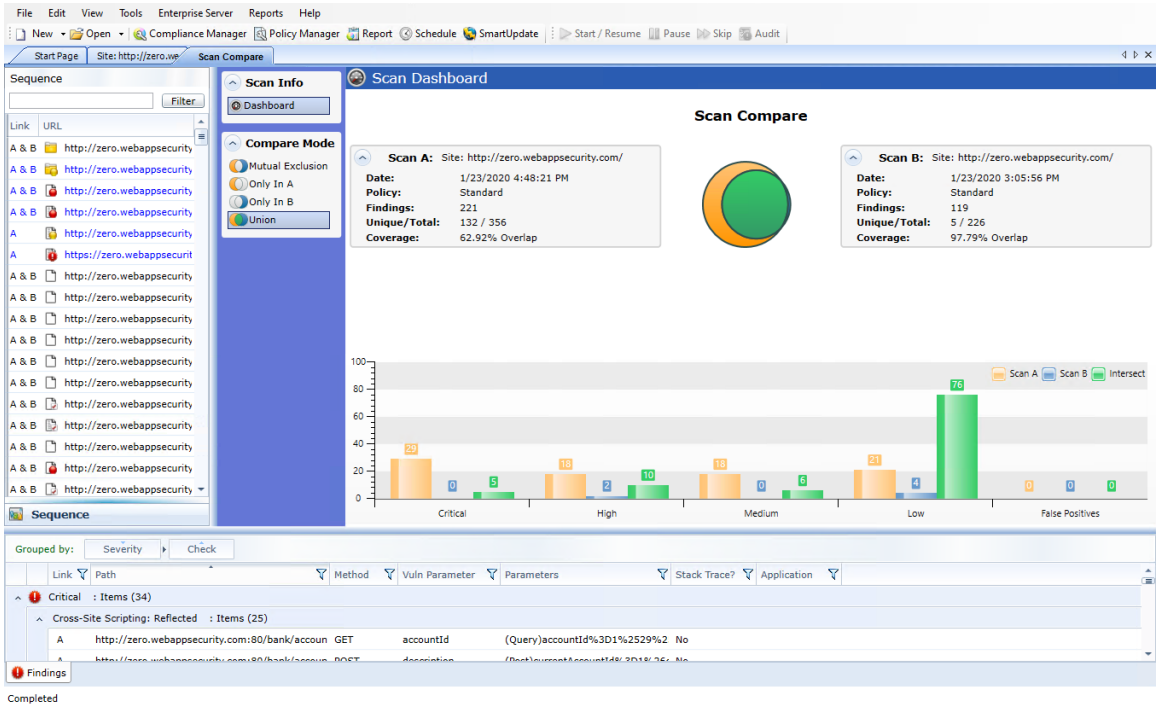
- From the Manage Scans page, select two scans and click **Compare**.
- From a tab containing an open scan (which will be Scan A in the comparison):
  a. Click **Compare**.
  b. Select a scan from the list on the Scan Comparison window. This scan will be Scan B in the comparison.
  c. Click **Compare**.

> **Note:** If the open scan is a "site retest" (resulting from **Rescan > Retest Vulnerabilities**), OpenText DAST 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**, OpenText DAST will compare "site retest - zero" with the parent scan "zero."

A warning message appears if the selected scans have different start URLs or used different scan policies, or if the scans are of a different type (such as a Basic Scan vs. a web service scan). You can choose to continue, or you can terminate the function.

You cannot conduct a comparison if either of the scans is currently running.

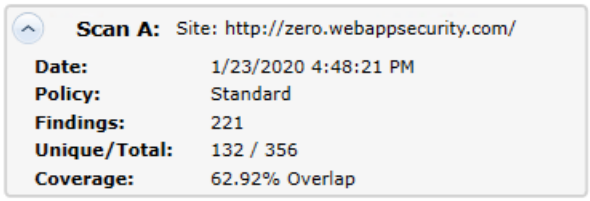
Scan Compare Image



Reviewing the Scan Dashboard

The Scan Dashboard displays the scan comparison results.

Scan descriptions



The Scan A and Scan B boxes provide the following information of the scans:

- **Scan A** or **Scan B**: Name of the scan.
- **Date**: Date and time the original scan was conducted.
- **Policy**: Policy used for the scan; see "OpenText DAST policies" on page 458 for more information.
- **Findings**: Total number of issues identified on the Findings tab, as well as false positives detected.
- **Unique/Total**: Number of unique sessions created for this scan (that is, the number of sessions that appear in this scan and not the other scan), compared to the total number of sessions for this scan.
- **Coverage**: Percentage of sessions that are common to both scans.

The Venn diagram

The Venn diagram depicts the session coverage of Scan A (represented by a yellow circle) and the session coverage of Scan B (represented by a blue circle). The intersection of the two sets is represented by the green overlap. (In prior releases, the Venn diagram represented the overlap of vulnerabilities.)

The Venn diagram is scaled to reflect the actual relationship between the sets.

Several examples of session coverage overlap are illustrated below.

				
No Intersection	50% Intersection	A Encompasses B	Most of A Intersects B	Complete Intersection

Vulnerabilities bar chart

In separate groupings for each vulnerability severity and for False Positives, the bottom of the Scan Dashboard displays a set of bar charts that show the number of vulnerabilities found in Scan A, in Scan B, and in their intersection (**Intersect**). The same color coding is used as in the Venn diagram. These bar charts do not change based on the selected **Compare Mode**.

Effect of scheme, host, and port differences on scan comparison

OpenText DAST does not ignore the scheme, host, and port when comparing scans from two duplicate sites that are hosted on different servers.

For example, the following site pairs would not be correlated in a scan comparison because of differences in scheme, host, or port:

- **Scheme**
  - Site A - http://zero.webappsecurity.com/
  - Site B - https://zero.webappsecurity.com/
- **Host**
  - Site A - http://dev.foo.com/index.html?par1=123&par2=123
  - Site B - http://qa.foo.com/index.html?par1=123&par2=123
- **Port**
  - Site A - http://zero.webappsecurity.com:80/
  - Site B - http://zero.webappsecurity.com:8080/

Compare modes

You can select one of the following options in the **Compare Mode** section to the left of the Scan Dashboard to display different data in the **Sequence** area in the left pane (the data in the Scan Dashboard is not affected):

- **Mutual Exclusion**: Lists sessions that appear in Scan A or Scan B, but not in both scans
- **Only In A**: Lists sessions that appear only in Scan A
- **Only in B**: Lists sessions that appear only in Scan B
- **Union** (the default): Lists sessions that appear in Scan A, Scan B, or both Scans A & B

Session filtering

The **Sequence** pane lists each session that matches the selected Compare Mode. An icon to the left of the URL indicates the severity of the vulnerability, if any, for that session. The severity icons are:

Critical	High	Medium	Low
			

At the top of the **Sequence** pane, you can specify a filter and click **Filter** to limit the set of displayed sessions in the following ways:

- You can enter the URL with only its starting characters, as a "starts with" match. Your entry must begin with the protocol (http:// or https://).
- You can search for an exact match by specifying the URL in quotes. Your entry must begin with the quotes and protocol ("http:// or "https://)
- You can use an asterisk (*) as a wildcard character at the beginning or end of the string you enter.
- You can use asterisks (*) at both the beginning and end of the string you enter, which requires matches to contain the string between the asterisks.
- You can enter a question mark (?) followed by a full query parameter string to find matches to that query parameter.

Using the Session Info panel

When you select a session in the **Sequence** pane, the **Session Info** panel opens below the **Compare Mode** options. With a session selected, you can select an option in the **Session Info** panel to display more details about that session to the right of the **Session Info** panel. If the session contains data for both scans, the data for some functions such as **Web Browser**, **HTTP Request**, and **Steps** are shown in a split view with Scan A on the left side and Scan B on the right side.

Note: The **Steps** option displays the path taken by OpenText DAST to arrive at the session selected in the **Sequence** 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. In a scan comparison, if any of the steps for the session are different between the scans, the **In Both** column is added to the **Steps** table (as the first column). A value of **Yes** in the column for a particular step indicates that the step is the same for that session for both scans A and B. A value of **No** in the column for a particular step indicates that the step is different for that session between scans A and B.

Using the Summary Pane to review vulnerability details

When comparing scans, the horizontal Summary pane at the bottom of the window provides a centralized table of vulnerable resources and enables you to quickly access vulnerability information. You can drag the horizontal divider above the table to show or hide more of the Summary pane.

The set of entries (rows) displayed in the **Findings** tab depends on the option selected for **Compare Mode**, as reflected in the **Link** column in the table.

Grouping and sorting vulnerabilities

For information on grouping and sorting vulnerabilities, see "Summary pane" on page 87 and "Using filters and groups in the Summary pane" on page 244.

Filtering vulnerabilities

You can click the filter icon ( ) at the right of any column heading to open a filter that enables you to choose a variety of conditions regarding that column that must be met in order for a vulnerability (row) to remain listed in the table after filtering. The available conditions include the full set of current values in the column, and you can also specify logical expressions regarding the content of that column.

For example, in the filter for the **Vuln Parameter** column, suppose you:

1. Leave the top set of check boxes as is.
2. Below the **Show rows with value that** text, select **Contains** from the drop-down menu.
3. Type **Id** in the text box below the drop-down menu.
4. Click **Filter**.

Then the table will show only rows that contain the text "Id" in the **Vuln Parameter** column. This would include rows for which the value of **Vuln Parameter** is **accountId** or **payeeId** or any other entry that includes "Id."

You can specify filters for multiple columns, one column at a time, and they will all be applied.

If a filter for a column has been specified, its icon becomes a darker blue than the icons for unused filters.

To quickly clear a filter, click **Clear Filter** while the filter is open to be specified.

Working with vulnerabilities

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.

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

See also

"Summary pane" on page 87

"Using filters and groups in the Summary pane" on page 244

Manage Scans

To manage scans:

1. On the **Start Page**, click **Manage Scans**.
   A list of scans appears in the right-hand pane of the **Start Page**.
   By default, OpenText DAST 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. For more information, see "Scan status" on page 202.
2. (Optional) To group scans into categories based on the column headings, drag the heading and drop it on the grouping area.
3. Use the toolbar to perform the tasks described in the following table.

To...	Then...
Search for a scan	Type the scan name or scan ID in the **Search** box. OpenText DAST filters the list of scans as you type. **Tip:** To clear the search criteria, click the clear search icon ().
Open one or more scans	Select one or more scans and click **Open** (or simply double-click an entry in the list). OpenText DAST loads the scan data and displays each scan on a separate tab.
Launch the Scan Wizard prepopulated with settings last used for the selected scan	Click **Rescan > Scan Again**.
Reuse a scan	Click **Rescan** and select the reuse option you want from the drop-down menu. For more information, see "Reusing scans" on page 232.
Rescan only those sessions that	Select a scan and click **Rescan >**

To...	Then...
contained vulnerabilities revealed during a previous scan	**Retest Vulnerabilities**.
Merge scans	Select two scans (using **Ctrl** + click), right-click, and select **Merge**. For more information, see "Incremental scanning" on page 233.
Rename a selected scan	Click **Rename**.
Delete the selected scan(s)	Click **Delete**.
Import a scan	Click **Import**.
Export a scan or scan details, or to export a scan to OpenText Application Security Center (Software Security Center), or to export protection rules to a web application firewall (WAF)	Click the drop-down button on **Export**.
Compare scans	Select two scans (using **Ctrl** + click) and click **Compare**.
Change the database connection settings for scan and report storage or for scan viewing or both **Note:** By default, OpenText DAST lists all scans that are saved in the local SQL Server Express Edition and in a configured SQL Server Standard Edition.	Click **Connections**. For more information, see "Application settings: Database" on page 435.
Update the display	Click **Refresh**.
Select which columns should be displayed	Click **Columns**. **Tip:** 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.

Note: You can also perform most of these functions by right-clicking an entry and selecting a command from the shortcut menu. In addition, you can also choose to generate a report. For more information, see "Generating a report" on page 260.

See also

"Managing scheduled scans" on the next page

"Start Page " on page 38

Schedule a scan

You can schedule a Basic Scan or an API Scan to occur at a date and time of your choosing.

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

Note: To access scheduled scans after they are complete, select the **Start Page** tab and click **Manage Scans**.

To schedule a scan:

1. Do one of the following:
   - Click the **Schedule** icon on the OpenText DAST toolbar.
   - Click **Manage Scheduled Scans** on the OpenText DAST **Start Page**.
2. When the Manage Scheduled Scans window appears, click **Add**.
3. In the **Type of Scan** group, choose one of the following:
   - **Web Site Scan**
   - **API Scan**
4. To conduct the scan one time only, select **Run Once** and then edit the **Start Date** and **Time**. If you click the drop-down arrow, you can use a calendar to select the date.
5. To scan a site periodically:
   a. Select **Recurring** (or **Recurrence Schedule**), then specify the start time and choose a frequency: **Daily**, **Weekly**, or **Monthly**.
   b. If you select **Weekly** or **Monthly**, provide the additional requested information.
6. Click **Next**.

See also

"Running a Basic Scan (website scan)" on page 171

"Using the API Scan Wizard" on page 137

"Configuring time interval for scheduled scan " below

Configuring time interval for scheduled scan

To configure when to run a scan or to set up recurring scans:

1. In the **Type of Scan** group, choose one of the following:
   - Basic Scan
   - API Scan
2. To conduct a scan now, select **Immediately**.
3. To conduct a one-time-only scan at a later date or time:
   a. Select **Run Once**.
   b. Modify the date and time when the scan should begin.

   Tip: Click the drop-down arrow to reveal a calendar for selecting the date.
4. To scan a site periodically:
   a. Select **Recurring**.
   b. Specify the time when the scan should start.
   c. Choose a frequency: Daily, Weekly, or Monthly.
5. Click **Next**.

See also

"Running a Basic Scan (website scan)" on page 171

"Using the API Scan Wizard" on page 137

Managing scheduled scans

You can instruct OpenText DAST 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 OpenText DAST (if necessary) and initiates the scan. It is not necessary for OpenText DAST 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 OpenText DAST Start page. To access scheduled scans after they are complete, select the Start page and click Manage Scans.

Accessing scheduled scans

To access a list of scheduled scans:

- On the **Start Page**, click **Manage Schedule**.

  A list of scans you previously scheduled appears in the right-hand pane of the **Start Page**.

  The current state of the scan is indicated in the Status column. For more information, see "Scheduled scan status" on page 217.

Deleting a scan

To delete a previously scheduled scan from the list:

- Select a scan and click **Delete**.

Editing scan settings

To edit settings for a scheduled scan:

- Select a scan and click **Edit**.

Running a scan immediately

To run a scan immediately, without waiting for the scheduled time:

- Select a 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.

Stopping a scheduled scan

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).

Scheduling a scan

To schedule a scan:

1. Click **Add**.
2. In the **Type of Scan** group, choose one of the following:
   - Basic Scan
   - Web Service Scan
3. 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.
4. Click **Next**.
5. Enter the settings for the type of scan you selected.
6. For website and web service scans only, you can elect to run a report at the conclusion of the scan:
   a. Select **Generate Reports** and click the **Select Reports** hyperlink.
   b. Continue with Selecting a Report (below).
7. 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 Scheduled Scan Report Wizard appears.

To select a report:

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 window appears.

Configuring report settings

To 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 then 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**.

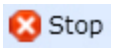
See also

"Start Page " on page 38

"Manage Scans" on page 210

"Scheduled scan status" on the next page

Stopping and restarting 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).

Scheduled scan status

The status of each scheduled scan appears in the **Last Run Status** column on the **Manage Schedule** pane. The possible statuses are defined in the following table.

Status	Definition
Failure	OpenText DAST was unable to perform the scan.
Success	The scan was conducted without error.
Not Yet Run	The scan is queued to run at the scheduled time, which has not yet occurred.
Skipped	The scheduled scan was not run because the service was down for some period of time.
Stopping	The user clicked the **Stop** button, but the scan has not yet stopped.
Stopped	The scan has been stopped by the user.
Running	The scheduled scan is in progress.
Running with Error	The scan could not stop; see log for further details.

Exporting a scan

Use the export scan function to save information collected during an OpenText DAST crawl or audit.

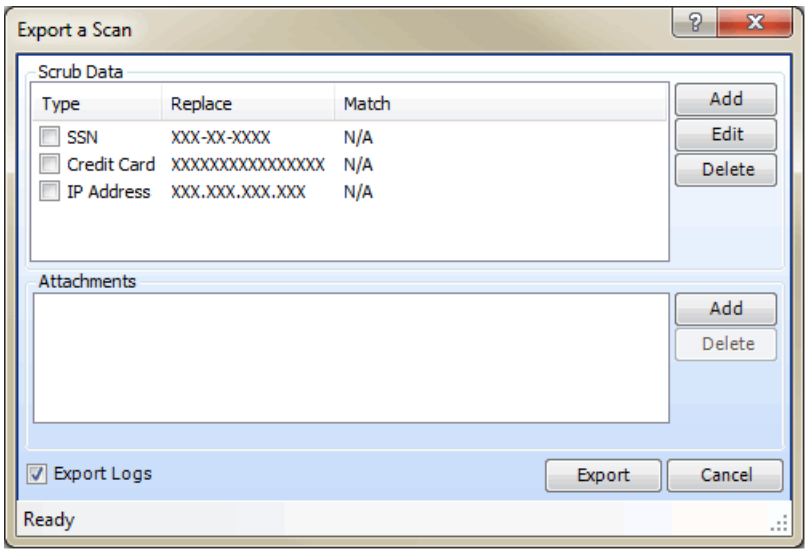
Note: When exporting to Application Security, after exporting to the .fpr format, you must manually upload the .fpr file to Application Security. OpenText does not support uploading OpenText DAST FPR artifacts to the same application version in Application Security.

Follow the steps below to export a scan.

1. Do one of the following:
   - Open a scan (or click a tab containing an open scan), click **File > Export** and select either **Scan** or **Scan to Application Security Center**.
   - 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 Application Security Center**.

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



2. The **Scrub Data** group contains, by default, three non-editable regular expression functions that will substitute **X**'s 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. This feature prevents any sensitive data from being included in the export.
3. 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 ellipsis 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**.
4. If you are exporting to Application Security, 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**.

See also

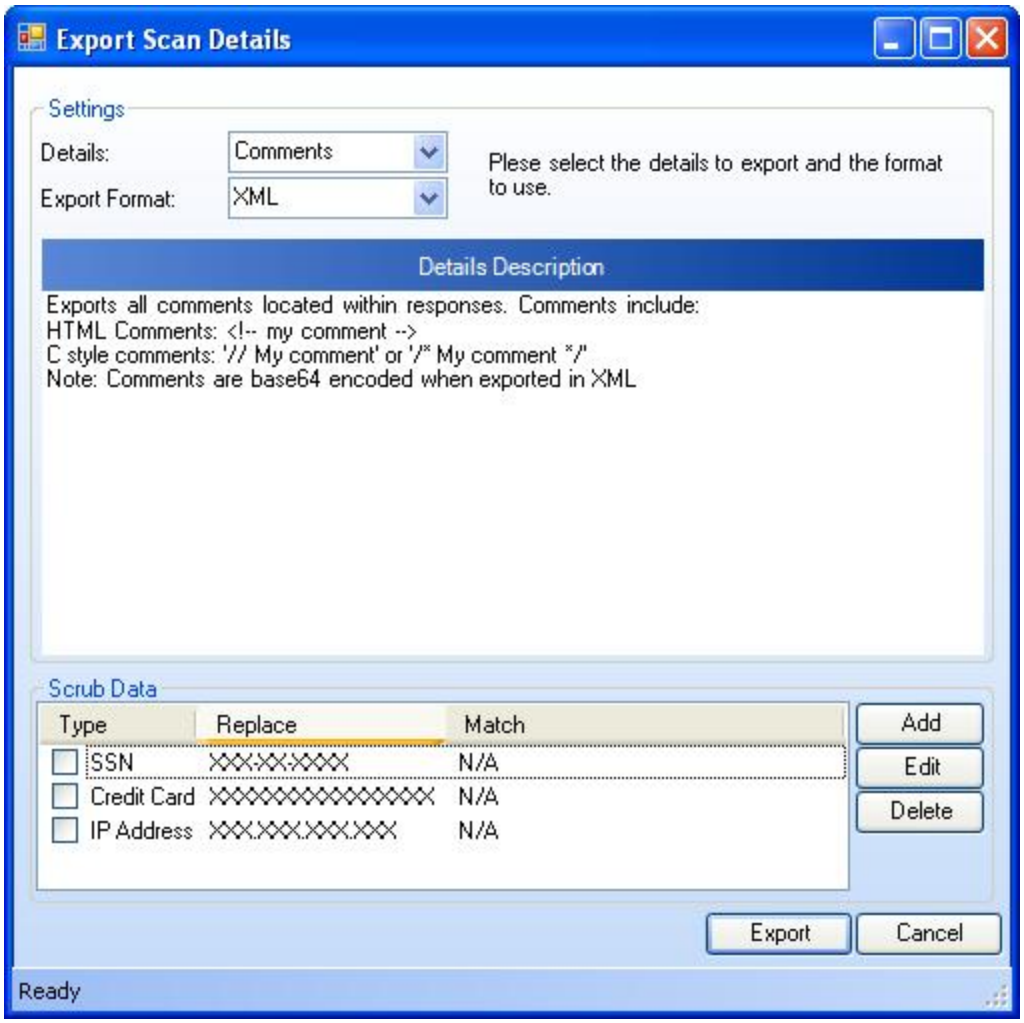
"Importing a scan" on page 223

"Exporting scan details" below

Exporting scan details

Use this function to save information collected during an OpenText DAST 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 options are as follows:
   - **Comments** – Exports all comments located within responses.

     Note: Comments are base64-encoded when exported in XML format.

   - **CycloneDX** – Exports a CycloneDX software bill of material (SBOM) for open source client-side libraries.

     Note: CycloneDX output is a JSON file. The **Export Format** list is not available for this output type.

     Tip: Application Security requires a ZIP file format when importing CycloneDX SBOM data. As a workaround, you must add the JSON file along with a file named `scan.info` to a ZIP file. The scan.info file must include the following line of text:

     ```
     engineType=WEBINSPECT
     ```

     Currently, on the ARTIFACT HISTORY page in Application Security, the engine **Type** will be identified as "Unknown."

   - **Emails** – Exports a list of all email addresses discovered during a scan.
   - **Full** – Exports session requests and responses (parsed and unparsed), as well as all session vulnerability information.
   - **Hidden Fields** – Exports all hidden fields located within responses.

     Note: Hidden fields are base64-encoded when exported in XML format.

   - **Offsite Links** – Exports all offsite links.
   - **Parameters** – Exports all sessions in the scan tree that have parameters.
   - **Requests** – Exports all URLs and the associated raw requests that were sent during the crawl.

     Note: Requests are base64-encoded when exported in XML format.

   - **Scripts** – Exports all scripts located within responses.

     Note: Comments are base64-encoded when exported in XML format.

   - **Sessions** – Exports all URLs and associated raw requests and responses.

     Note: Requests and responses are base64-encoded when exported in XML format.

   - **Set Cookies** – Exports all cookies that were set by the server.

Note: Cookies are base64-encoded when exported in XML format.

- **URLs** – Exports all URLs that were crawled.
- **Vulnerabilities** – Exports vulnerabilities along with relevant contextual information.
- **Web Crawl Dump** – Dumps the response body of each crawled session to the file system with the original file name and directory structure.
- **Site Tree Dump** – Dumps the response body of each crawled session to the file system with the original file name and directory structure.
- **Web Forms** – Exports all forms located within responses.

Note: Not all choices are available for a web service scan.

4. From the **Export Format** list, choose an output format. Options are **Text** and **XML**. However, some types of details are available in one format only.
5. The **Scrub Data** group contains, by default, three non-editable regular expression functions that will substitute X's for each digit in a string formatted as a Social Security number, credit card number, or an IP address. To include this search-and-replace function for a data type, select its associated check box. This feature prevents any sensitive data from being included in the export.
6. 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 ellipsis 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, specify a name and location for the exported file and click **Save**.

See also

"Exporting a scan" on page 217

Export scan to Application Security Center

This feature enables you to export the results of an OpenText DAST scan in a format (.fpr format) that can be consumed by Application Security.

Note: After exporting to the .fpr format, you must manually upload the .fpr file to Application Security. OpenText does not support uploading OpenText DAST FPR artifacts to the same application version in Application Security.

1. Do one of the following:
   - Open a scan (or click a tab containing an open scan) and click **File > Export > Scan to Application Security Center**.
   - On the Manage Scans pane of the Start page, select a scan, click the drop-down arrow on the **Export** button and select **Export Scan to Application Security Center**.

   The Export Scan to Application Security Center window appears.
2. The **Scrub Data** group contains, by default, three non-editable regular expression functions that will substitute **X**'s 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. This feature prevents any sensitive data from being included in the export.
3. 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 ellipsis 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**.
4. Click **Export**.
5. Using the standard file-selection window, select a location and click **Save**.

Known issue with uploading FPRs from paused scans

The following scenario causes a known issue when manually exporting FPRs from OpenText DAST to Application Security:

1. The scan is started in OpenText DAST.
2. The scan is paused.
3. The scan is exported to FPR and then uploaded to Application Security.
4. The scan is restarted in OpenText DAST.
5. The scan is completed with additional vulnerabilities found in the application.
6. The scan is exported to FPR and uploaded to Application Security.

When uploading an FPR, Application Security uses the scan creation time in the FPR to determine whether the file already exists. In this scenario, both the partial scan and the

completed scan have the same creation time. Because the partial scan already exists in Application Security, the completed scan is not uploaded. The additional vulnerabilities that were found after the scan resumed will not appear in Application Security.

As a workaround, you must delete the partial FPR from Application Security before uploading the completed results.

Exporting protection rules to Web Application Firewall (WAF)

To generate and save a full export (.xml) file based on vulnerabilities detected by OpenText DAST during a scan of your web application:

1. Open the scan of interest (or click a tab containing an open scan) and click **File > Export > Protection Rules to Web Application Firewall**.
2. Specify the scrub data types in the same way as for the **File > Export > Scan** option. 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. This feature prevents any sensitive data from being included in the export.
3. Click **Export**.
4. Specify the path and filename to which you want to save the exported data and click **Save**.

   A full export (.xml) file is saved as you specified.

Importing a scan

To import a scan:

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 OpenText DAST versions beginning with 7.0.
   - SPA files (*.spa) - scan files created by versions of OpenText DAST prior to version 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:\Users\<username>\AppData\HP\HP WebInspect\ScanData\Imports\<DirectoryName>\<filename>`, where DirectoryName is the ID number of the exported or imported scan.

Important! If you see the error message "Unsupported macros have been removed from the scan settings," then you must record new macros using the latest Web Macro Recorder available.

See also

"Exporting a scan" on page 217

Selecting a scan to import suppressed findings

Use the **Select a Scan to Import Suppressed Findings** dialog to choose one or more scans from which to import suppressed findings into your current scan.

To import suppressed findings:

1. Select the checkbox(es) for the scan or scans from which you want to import suppressed findings, and click **OK**.

   The Importing Suppressed Findings window appears, displaying the progress of the import.
2. When the import is complete, do one of the following:
   - Click **Details** to view a log file for the import.
   - Click **Close** to view the false positive(s) in the Scan False Positives window.

Importing legacy web service scans

OpenText DAST 10.00 and later offer minimal support for web service scans that were created with versions of OpenText DAST earlier than 9.00. 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 "Vulnerability Review" feature is turned off.
- All reports should work as in previous OpenText DAST releases.
- The Scan view should render in "ReadOnly" mode, which disables the **Start**, **Audit**, and **Current Settings** buttons.

OpenText recommends that you rescan your web service.

Importing and exporting scan settings

If you require different settings for different scan actions, you can save your settings in an XML file and load them when needed. You can also reload the OpenText DAST factory default settings. You can perform these tasks in the Default Settings window

Tip: You can also create, edit, delete, import, and export scan settings files from the Manage Settings window. Click **Edit** and select **Manage Settings**.

To import, export, or restore settings:

1. Click **Edit > Default Settings**.
   The Default Settings window appears.
2. Continue according to the following table.

To...	Then...
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**.
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**.
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**.

Chapter 5: Using OpenText DAST features

This chapter describes certain tools available in OpenText DAST, such as the Server Profiler and Web Macro Recorder tools. It also describes how to inspect the scan results and work with vulnerabilities discovered during the scan. It describes using the OpenText DAST API, Regular Expressions, and the OpenText DAST policies. This chapter also includes information about Compliance Templates and the reporting capabilities of OpenText DAST.

For more information about all tools available in OpenText DAST, see the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Retesting and rescanning

OpenText DAST offers several methods for retesting and rescanning discovered vulnerabilities. You may:

- Retest an individual vulnerability, all vulnerabilities, or all vulnerabilities with a specific severity. For more information, see "Retesting vulnerabilities" below.
- Rescan the entire site. For more information, see "Rescanning a site" on page 231.
- Reuse data from a previous scan to assist a new scan. For more information, see "Reusing scans" on page 232 and "Incremental scanning" on page 233.

Retesting vulnerabilities

After you conduct a scan and report discovered vulnerabilities, developers may correct their code and update the site. Afterward, you can open the original scan and conduct a retest scan to verify the fix for:

- A selected vulnerability
- All vulnerabilities
- All vulnerabilities with a specific severity

OpenText DAST starts a new scan to determine whether the issue or issues have been fixed. The retest scan prefixes "`retest:`" to the original scan name so that you can easily discern the retest scan from the original scan.

During the retest scan, the vulnerabilities that are queued for retesting are listed on the Findings tab in the Summary pane, along with a Retest Status column that indicates the results of the retest.

Important! OpenText does not recommend retesting vulnerabilities in scans created using earlier versions of OpenText DAST. While retesting scans from earlier versions may work in many instances, it is not always reliable because individual checks may not flag the same vulnerability during a retest. Failure of a check to flag the same vulnerability while retesting a scan from an earlier version of OpenText DAST may not mean the vulnerability has been remediated.

Understanding the retest status

The following table describes the values that may appear in the Retest Status column.

Status	Description
Processing	The vulnerability is currently being retested. This is a temporary status that will be replaced with a final status when the retest is complete.
Detected	The vulnerability was reproduced in the retest scan.
Not Detected, Possible Correlation Failure	A vulnerability with the same check ID was detected during the retest scan, but the correlation did not match the finding that was being retested. **Note:** Correlation refers to how OpenText DAST uniquely identifies a vulnerability using the same parameter or location.
Not Detected	The vulnerability does not exist in the parameter or location being tested.
Not Supported	The vulnerability was not retested. Retesting is not supported for the specific vulnerability. For more information, see "Recommendation for failed and not supported vulnerabilities" on the next page.
Failed	Retesting failed for the specific vulnerability. You may also see the following failed statuses that provide a reason for failure: • Failed, Trigger Session Not Found • Failed, Trigger Session Response Missing • Failed, Trigger Session Status Code Different For more information, see "Recommendation for failed and not supported vulnerabilities" on the next page.
Dependency Failed	Retesting failed to complete because a dependency that existed in the original scan could not be duplicated in the verification scan.

Recommendation for failed and not supported vulnerabilities

For vulnerabilities with a Retest Status of "Failed" or "Not Supported," OpenText recommends that you conduct a reuse remediation scan or a new scan. For more information about reuse remediation scans, see "Reusing scans" on page 232.

Retesting all vulnerabilities

To retest all vulnerabilities in a scan:

- Do one of the following:
  - In the **Manage Scans** list, right-click a scan and select **Rescan > Retest Vulnerabilities > Retest All**.
  - In the Scan menu of an open scan, click the **Rescan** drop-down list and select **Retest Vulnerabilities > Retest All**.
  - In the **Findings** tab on the Summary pane of an open scan, right-click a vulnerability and select **Retest > Retest All**.

  The retest scan starts with "`retest:`" prefixed to the original scan name.

Retesting all vulnerabilities with a specific severity

To retest all vulnerabilities with a specific severity in a scan:

1. Do one of the following:
   - In the **Manage Scans** list, right-click a scan and select **Rescan > Retest Vulnerabilities > Retest by Severity**.
   - In the Scan menu of an open scan, click the **Rescan** drop-down list and select **Retest Vulnerabilities > Retest by Severity**.
   - In the **Findings** tab on the Summary pane of an open scan, right-click a vulnerability and select **Retest > Retest by Severity**.
2. Select the specific severity (**Critical**, **High**, **Medium**, or **Low**).

   Note: If a severity is unavailable in the context menu, then the scan does not have vulnerabilities of that severity.

   The retest scan starts with "`retest:`" prefixed to the original scan name.

Retesting selected vulnerabilities

To retest one or more selected vulnerabilities:

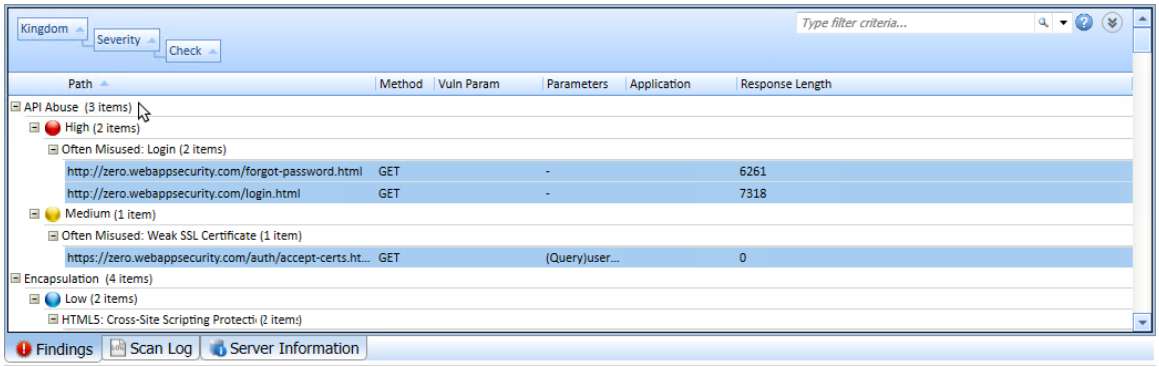
1. In the **Findings** tab on the Summary pane of an open scan, do one of the following:
   - To retest one vulnerability, right-click the vulnerability.
   - To retest multiple vulnerabilities, press `CTRL` and click the vulnerabilities to select them, and then right-click.
2. Select **Retest > Retest Selected**.
   The retest scan starts with "`retest:`" prefixed to the original scan name.

Retesting grouped categories

If you have findings grouped into categories, you can select a group and retest all items in that category.

To retest a group:

1. Select the group to retest.
   All findings in the group are selected. In the following image, for example, findings are grouped by Kingdom, then Severity, and then Check. The API Abuse group is selected, so all findings in that category are selected.



2. Right-click, and then select **Retest > Retest Selected**.

Tip: You can right-click the group to select all findings in the category and display the context menu in a single action.

For more information about groups, see "Using filters and groups in the Summary pane" on page 244.

Retesting a retest scan

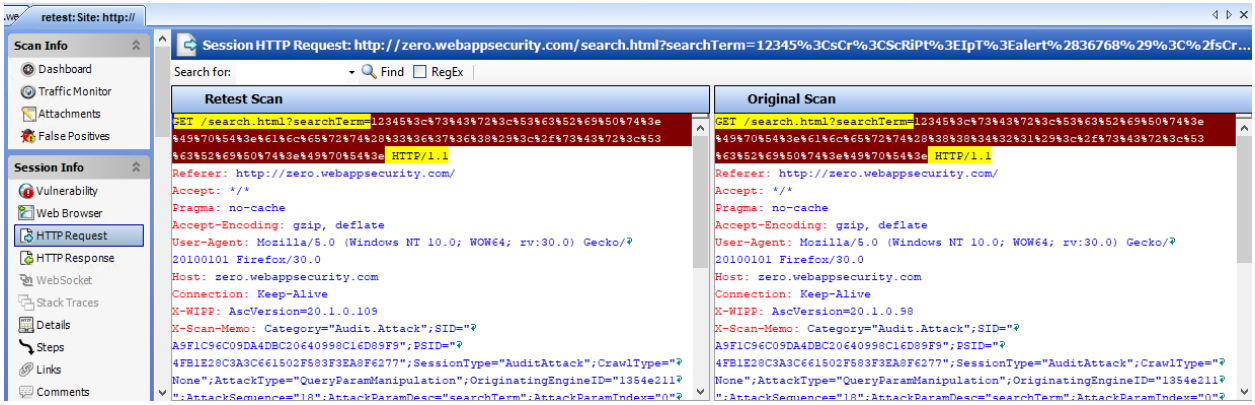
You can retest the findings in a retest scan in the same way you retest an original scan. However, you can only retest findings with a Retest Status of Detected. Findings with other Retest Statuses will not be retested.

Retest scan log

If you retest a large number of findings in a scan, you can view a snapshot of the results in the Scan Log tab for the retest scan.

Comparison views

When you select a vulnerability in a retest scan, you can view certain data from both scans in a dual-pane view. Select **HTTP Request**, **HTTP Response**, or **Steps** to display a dual pane view comparing the retest scan to the original scan. If the original scan is not available, only the data for the retest scan is displayed.



To search for data in the HTTP Request and HTTP Response views:

1. Type the search term in the **Search for** field.
2. Optionally, to use regular expressions in the search criteria, select the **RegEx** option.
3. Click **Find**.

   If the data is found, it is highlighted in both Retest Scan and Original Scan.

For more information, see "HTTP Request" on page 73, "HTTP Response" on page 74, and "Steps" on page 74.

Keeping or deleting a retest scan

When you close an open scan, OpenText DAST detects whether it is a retest scan. If the following conditions are met, you will be prompted about keeping the scan:

- It is a retest scan.
- The parent scan exists in the scan database.

- You have not been previously prompted for the scan.

When these conditions are met, a prompt asks **Do you want to keep the scan "retest:<*ScanName*>"?** If you close multiple tabs for retest scans meeting these conditions, a prompt appears for each retest scan.

Do one of the following:

- To keep the retest scan, click **Yes**.

  The scan is saved and added to the Recently Opened Scans list. Additionally, the scan's settings are flagged to prevent the prompt from being shown again. This flag will be preserved even if the scan is exported and imported into another scan database.
- To delete the retest scan, click **No**.

  The Deleting Scans window appears. When the scan is deleted, click **Done**.

Rescanning a site

The Rescan feature enables 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.

There are also two options for reusing a scan: Reuse Incremental and Reuse Remediation. For more information, see "Reusing scans" on the next page.

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 OpenText DAST 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**.

Reusing scans

Reusing a scan uses data from a previous scan to assist a new scan. Two scans are involved when conducting a reuse scan:

- The reuse scan is the new scan being conducted.
- The source or baseline scan is the scan from which data is used to reduce the work and time needed to complete a reuse scan.

Reuse options

Four options for scan reuse are available:

- **Reuse Incremental** – find new attack surface. This scan performs a normal crawl and compares each session to the baseline scan. Only new sessions that did not exist in the baseline scan are audited. For more information, see "Incremental scanning" on the next page.
- **Reuse Remediation** – look for vulnerabilities that were found in the baseline scan. This scan creates a policy that includes only those checks that flagged in the baseline scan, and audits the site again using this custom policy. Therefore, this scan looks at only the checks that flagged in the baseline scan.

Difference between remediation scans and retest vulnerability

Remediation scans apply a reduced policy that is derived directly from the flagged vulnerabilities in the baseline scan to all sessions in the remediation scan, rather than to just the sessions that were vulnerable in the baseline scan.

For example, a baseline scan found cross-site scripting (XSS) on session A but not session B. Subsequently, XSS was fixed on session A, but created on session B. Using the Retest Vulnerabilities option will not find the vulnerability on session B, but a remediation scan will find it. Therefore, a remediation scan will evaluate all of the known attack surface area for previously found vulnerabilities.

Guidelines for reusing scans

Follow these guidelines when reusing scans:

- The baseline scan must be available on the machine where the reuse scan is executed.
- The baseline scan does not need to be in the same database as the reuse scan.

Reusing a scan

To reuse a scan:

1. Do one of the following:
   - From an open scan, click **Rescan** and select the reuse option you want from the drop-down menu.
   - On the Manage Scans page, right-click a scan, click **Rescan**, and then select the reuse option you want from the menu.
   - On the Manage Scans page, select a scan, click **Rescan** and select the reuse option you want from the drop-down menu.

   For information about the rescan options, see "Reuse options" on the previous page.
2. Using the Scan Wizard, you may optionally modify the settings that were used for the original scan.

   Tip: For incremental scans, it might be beneficial to change settings to discover new attack surface. However, changing settings is not recommended for remediation scans.

   Note: By default, the type of reuse scan you selected is prepended to the baseline scan name and a -1 is appended to the end.
3. On the last step of the Scan Wizard, click **Scan**.

See also

"Incremental scanning" below

Incremental scanning

Incremental scanning provides a way for you to find and audit the areas of your web application that change over time, while keeping all findings in a single scan. This involves performing incremental scans and merging these scans back into the baseline scan. For more information about incremental scans and baseline scans, see "Reusing scans" on the previous page.

Merging baseline and incremental scans

You can merge the baseline scan and the incremental scan into a single scan. Then you can use the attack surface of the combined scans for future incremental scans.

After conducting an incremental scan, if you select the incremental scan and the baseline scan and then right click, you will see a Merge option.

Important! You must click the baseline scan from which the incremental scan was derived to see the Merge option enabled.

When you click Merge, the incremental scan is merged into the baseline scan. The baseline scan now contains the union of the 2 scans. After merging, the resulting scan becomes the new baseline scan. You can continuously perform incremental-merge-incremental-merge indefinitely to create a process for continuous or deferred auditing.

For more information, see "Incremental with continuous audit" below and "Incremental with deferred audit" below.

To merge scans:

1. In the Manage Scans page, select the baseline scan and the incremental scan.
2. Right-click and select **Merge**.

Log entries, including the baseline and incremental scan IDs, are written to the scan log when scans are merged.

Incremental with continuous audit

With incremental scanning, you can put in place a process for continuous audit. This process would be as follows:

1. Create a baseline scan.
2. When an incremental scan is needed:
   a. Create an incremental audit scan from the baseline scan. During this scan, new surface is audited.
   b. Merge the incremental scan with the baseline scan. The merged scan becomes the new baseline scan. For more information, see "Merging baseline and incremental scans" on the previous page.
   c. Delete the incremental scan.
   d. Return to Step 2.

Incremental with deferred audit

With incremental scanning, you can put in place a process for deferred audit. This process would be as follows:

1. Create a baseline scan.
2. When a new incremental scan is needed:
   a. Create an incremental crawl-only scan from the baseline scan.
   b. Merge the incremental scan with the baseline scan. The merged scan becomes the new baseline scan. For more information, see "Merging baseline and incremental scans" on the previous page.
   c. Delete the incremental scan.
   d. If new attack surface is found, resume the baseline audit and audit the new surface.
   e. Return to Step 2.

See also

"Reusing scans" on page 232

Using macros

A macro is a recording of the events that occur when you access and log in to a website. You can subsequently instruct OpenText DAST to begin a scan using this recording. You can use either the Session-based Web Macro Recorder tool or the Event-based Web Macro Recorder tool to record login macros, or you can create them in the Basic Scan or Guided Scan wizards. Macros that are created in a Basic Scan or a Guided Scan can be used in either type of scan.

There are two types of macros:

- A login macro is a recording of the events that occur when you access and log in to a website using a Web Macro Recorder tool. You can subsequently instruct OpenText DAST to begin a scan using this recording.

  If **Enable macro validation** is selected in Scan Settings: Authentication for scans that use a login macro, OpenText DAST tests the login macro at the start of the scan to ensure that the log in is successful. If the macro is invalid and fails to log in to the application, the scan stops and an error message is written in the scan log file. For more information and troubleshooting tips, see "Testing login macros" on page 494.

  Note: Macro testing is not supported for macros containing two-factor authentication.

- A workflow macro is a recording of HTTP events that occur as you navigate through a website using a Web Macro Recorder tool. OpenText DAST audits only those URLs included in the macro that you previously recorded and does not follow any hyperlinks encountered during the audit. Supported macros are `.webmacro` files, Burp Proxy captures, and `.har` files.

Any activity you record in a macro will override the scan 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, OpenText DAST will ignore the exclusion when it crawls and audits the site.

Note: When you play a macro, OpenText DAST will not send any cookie headers that may have been incorporated in the recorded macro. Macros that were recorded in a Basic Scan or a Guided Scan can be used in either type of scan.

See also

"Scan settings: Authentication" on page 393

"Running a Guided Scan" on page 96

"Running a Basic Scan (website scan)" on page 171

"Selecting a workflow macro " on the next page

"Using a Web Macro Recorder" on the next page

Selecting a workflow macro

When conducting a Workflow-driven Scan, you can select or create one or more macros that will be used to navigate your website.

- **Record** - opens the Web Macro Recorder, allowing you to create a macro
- **Edit** - opens the Web Macro Recorder and loads the selected macro
- **Remove** - removes the selected macro (but does not delete it from your disk)
- **Import** - opens a standard file-selection window, allowing you to select a previously recorded `.webmacro` file, Burp Proxy captures, or `.har` file.

  Important! If you use a login macro in conjunction with a workflow macro or startup macro or both, all macros must be of the same type: all `.webmacro` files or all Burp Proxy captures or all `.har` files. You cannot use different types of macros in the same scan.

- **Export** - opens a standard file-selection window, allowing you to save a recorded macro

Once a macro is selected or recorded, you may optionally specify allowed hosts.

See also

"Using macros" on the previous page

Using a Web Macro Recorder

OpenText DAST includes two versions of Web Macro Recorder tools:

- Event-based Web Macro Recorder
- Session-based Web Macro Recorder

The Web Macro Recorder tools can be launched in several ways–while configuring a Guided Scan or a Basic Scan, or outside of either scan in what is known as “stand-alone” mode. For more information, see the Web Macro Recorder help or the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Event-based Web Macro Recorder

OpenText DAST includes two Event-based Web Macro Recorder tools: one for login macros and one for workflow macros. In this document, these two tools are referred to generally as "Web Macro Recorder" except for specific login-related and workflow-related content.

The Event-based Web Macro Recorder tool was designed with TruClient technology. It uses event-based functionality and Firefox browser technology to record and play macros.

Session-based Web Macro Recorder

OpenText DAST includes Session-based Web Macro Recorder tools: one for login macros and one for workflow macros. In this document, these two tools are referred to generally as "Session-based Web Macro Recorder" except for specific login-related and workflow-related content.

The Session-based Web Macro Recorder uses Internet Explorer browser technology (also referred to here as IE technology) to record and play macros.

See also

"Using macros" on page 235

Traffic Monitor (Traffic Viewer)

OpenText DAST normally displays in the navigation pane only the hierarchical structure of the website or web service, plus those sessions in which a vulnerability was discovered. The Traffic Monitor or Traffic Viewer enables you to display and review every HTTP request sent by OpenText DAST and the associated HTTP response received from the web server.

The Traffic Monitor or Traffic Viewer is not available if Traffic Monitor Logging was not enabled prior to conducting the scan. You can enable the feature in the default settings (click **Edit > Default Settings > Settings > General**) or when you start a scan through the Scan Wizard (by selecting **Enable Traffic Monitor** on the Detailed Scan Configuration window under Settings).

Traffic session data in Traffic Viewer

The original Traffic Monitor has been converted into a standalone Traffic Viewer tool that incorporates functionality from both the original Traffic Monitor and the WebProxy tool. Traffic session files for the standalone Traffic Viewer use a different format than the Traffic Monitor. For more information about the standalone Traffic Viewer tool, refer to the Traffic Viewer tool online help or the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Viewing traffic in the Traffic Viewer

To view the traffic session data in the Traffic Viewer:

- In the Scan Info panel for an open scan, click **Traffic Monitor**.
  The Traffic Viewer tool opens with the traffic session data in view.

See also

"Scan Info panel" on page 58

Server Profiler

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

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

Similarly, your settings may specify that OpenText DAST should not conduct "file-not-found" detection. This process is useful for websites 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 OpenText DAST setting to accommodate this feature.

The Server Profiler can be selected during a Guided Scan, or enabled in the Application settings. For specific information, see "Application settings: Server Profiler" on page 439.

Launching Server Profiler as a tool

To launch the profiler as a tool, outside of a scan wizard:

1. Click the OpenText DAST **Tools** menu and select **ServerProfiler**.
2. In the **URL** box, enter or select a URL or IP address.
3. (Optional) If necessary, modify the **Sample Size**. Large websites may require more than the default number of sessions to sufficiently analyze the requirements.
4. Click **Analyze**.
   The Profiler returns a list of suggestions (or a statement that no modifications are necessary).
5. To reject a suggestion, clear its associated check box.
6. For suggestions that require user input, provide the requested information.
7. (Optional) To save the modified settings to a file:
   a. Click **Save Settings**.
   b. Using a standard file-selection window, save the settings to a file in your Settings directory.

Invoke Server Profiler when starting a scan

To launch the profiler when beginning a scan:

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

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

Inspecting the results

As soon as you start a scan, OpenText DAST begins scanning 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 **Findings** tab in the summary pane. For more information, see "Navigation pane" on page 46 and "Findings tab" on page 88.

Note: For Web Services and API scans, the Site tree is populated with icons depicting the operations and parameters from the Web Services Definition Language (WSDL) document or the API definition file.

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. For more information, see "Information pane" on page 57.

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**. For more information, see "Session Info panel" on page 69.

Working with one or more vulnerabilities

If you right-click one or more vulnerabilities in the summary pane, a shortcut menu enables 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** - Copies the item to a CSV file.
- **View in Browser** - Available if one vulnerability is selected; renders the HTTP response in a browser.
- **Filter by Current Value** - Available if one vulnerability is selected; 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: 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 244.

- **Change Severity** - Enables you to change the severity level.
- **Edit Vulnerability** - Available if one vulnerability is selected; displays the Edit Vulnerabilities dialog, allowing you to modify various vulnerability characteristics. For more information, see "Editing vulnerabilities" on page 251.
- **Rollup Vulnerabilities** - Available if multiple vulnerabilities are selected; enables you to roll up the selected vulnerabilities into a single instance that is prefixed with the tag “[Rollup]” in OpenText DAST and reports. For more information, see "Vulnerability rollup" on page 253.

  Note: If you have selected a rolled up vulnerability, this menu option is **Undo Rollup Vulnerabilities**.

- **Retest** - Performs a retest of one or more selected findings, all findings, or findings of a specific severity. For more information, see "Retesting vulnerabilities" on page 226.
- **Mark as** - Flags the vulnerability as either a false positive (and enables you to add a description) or as ignored. In both cases, the vulnerability is removed from the list. You can view a list of all false positive and ignored vulnerabilities by selecting **Suppressed Findings** in the Scan Info panel.

  Note: You can change a suppressed finding back to a vulnerability. See "Suppressed findings" on page 67 for details.

- **Send to** - Converts the vulnerability to a defect and adds it to the OpenText Application Lifecycle Management (ALM) 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 sessions" on page 258 for details.

- **Crawl** - Available if one vulnerability is selected; re-crawls the selected URL.
- **Tools** - Available if one vulnerability is selected; presents a submenu of available tools.
- **Attachments** - Available if one vulnerability is selected; enables 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.

Working with a Group

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

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

Understanding the Severity

The relative severity of a vulnerability listed in the summary pane is identified by its associated icon, as described in the following table.

Icon	Description
Critical	A vulnerability wherein an attacker might have the ability to execute commands on the server or retrieve and modify private information.
High	Generally, the ability to view source code, files out of the web root, and sensitive error messages.
Medium	Indicates non-HTML errors or issues that could be sensitive.
Low	Interesting issues, or issues that could potentially become higher

Icon	Description
	ones.
Information	An interesting point in the site, or detection of certain applications or web servers.
Best Practice	Issues related to commonly accepted best practices for web development that may indicate overall site quality and site development security practices (or lack thereof).

Working in the Navigation pane

You can also select an object or session in the navigation pane and investigate the session using the options available on the Session Info panel. For more information, see "Navigation pane" on page 46 and "Session Info panel" on page 69.

See also

"Retesting and rescanning" on page 226

"Auditing web services" on page 247

"Editing vulnerabilities" on page 251

"OpenText DAST user interface" on page 36

"Recovering deleted sessions" on page 258

Client-side library analysis

The hacker-level insights check has been enhanced to include information from the National Vulnerability Database (NVD) as well as Debricked health metrics.

Important! The number of findings for hacker-level insights in the Scan Dashboard may not match the number of URLs listed in the Summary Pane. Currently, hacker-level insight findings are not validated as exploitable and are not divided into separate vulnerabilities in the Scan Dashboard.

NVD information

If you select a policy that has the **Hacker Level Insights (HLI) Detected Libraries** check enabled, and a vulnerable library is detected on the client side, information from a local copy of the NVD about common vulnerabilities and exposures (CVE) will be included in the vulnerability summary.

Note: The NVD is shipped with the OpenText DAST installer. It is updated once per release, and is not updated between releases.

You can learn more about the National Vulnerability Database (NVD) at https://nvd.nist.gov/.

Debricked health metrics

If the detected library is open source, and you have a subscription to Debricked and have configured OpenText DAST with your Debricked access token, then information about the library contributors, popularity, and security will be retrieved from the Debricked database and included in the vulnerability summary. Access to Debricked is configured by way of the `WIConfig` program. For more information, see the *OpenText™ Dynamic Application Security Testing Installation Guide*.

A Debricked configuration also extends the local NVD and includes the newest CVEs. If there are no records for a CVE inside the local NVD, then data about the CVE and its description will be obtained from the Debricked database.

The Debricked information may also include correlated GitHub Security Advisory (GHSA) information for open source projects.

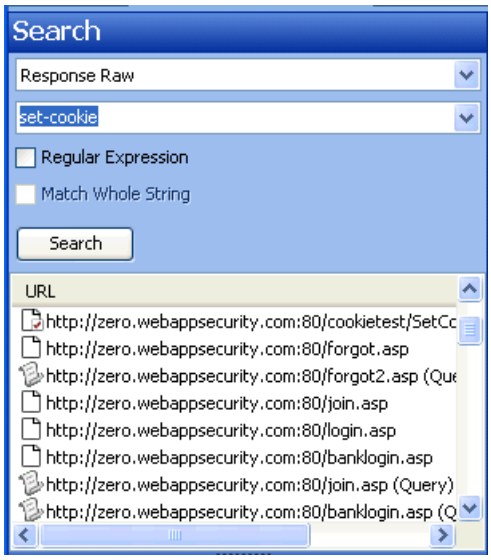
You can learn more about the Debricked health metrics at https://portal.debricked.com/project-health-45. You can learn more about GitHub Security Advisories at https://docs.github.com/.

Debricked content contingent upon access

If the Debricked service is down or unreachable for any reason at the start of a scan, the scan will continue. However, if access to the Debricked service has not been established upon scan completion, then Debricked information will not be included in the scan results.

Search view

The Search view enables you to search across all sessions for various HTTP message components. For example, if you select **Response Raw** from the drop-down and specify **set-cookie** as the search string, OpenText DAST 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). For more information, see "Navigation pane" on page 46.

   If all buttons are not displayed, click the **Configure Buttons** drop-down at the bottom of the button list and select **Show More Buttons**.
2. From the top-most list, select an area to search.
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. For more information, see "Regular expressions" on page 310.
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**.

See also

"OpenText DAST user interface" on page 36

Using filters and groups in the Summary pane

This topic describes how to use 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.

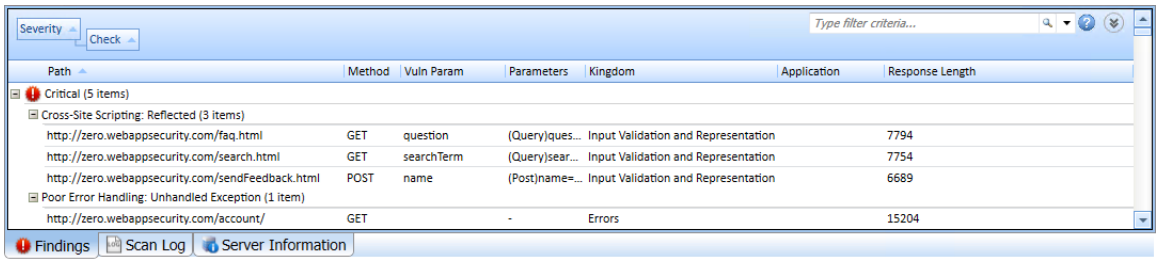
- Right-click a value in any column and select **Filter by Current Value** from the shortcut menu.

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

No filters

The following example shows unfiltered items on the **Findings** tab.

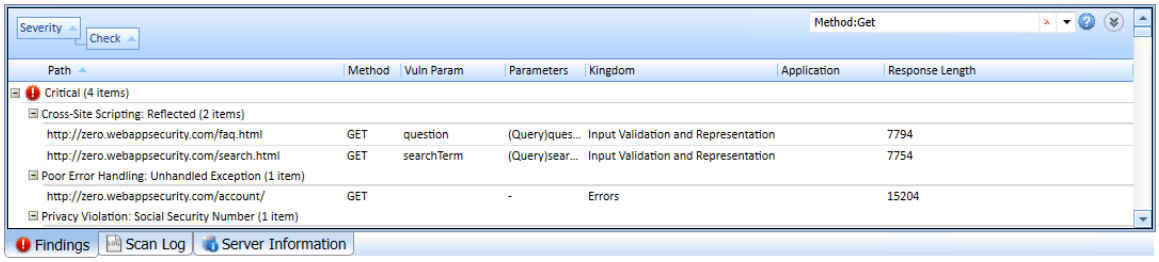
Summary Pane with No Filters Image



Filtered by Method:Get

The following example is rendered after entering "Method:Get" in the filter criteria box.

Summary Pane with Filters Image



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.

Specifying multiple filters

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

Filter criteria

You can enter the following identifiers:

- application - Application or framework in which the vulnerability is found
- check - Check name
- checkid - Check ID number from SecureBase
- 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
- cwe - Common Weakness Enumeration (CWE) ID
- duplicates - Duplicates detected by OpenText DAST Agent
- 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
- kingdom - Value from Seven Pernicious Kingdoms (for more information, see "Application settings: General" on page 431)
- 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)
- pendstatus - Status if the scan were to be published to Application Security
- rawrp - Raw HTTP response
- rawrq - Raw HTTP request
- responselength - Response size in bytes for the vulnerable session
- reteststatus - Retest Status values (for a list of values, see "Retesting vulnerabilities" on page 226)
- sessiondataid - Session data identifier (right-click on a session in the Navigation pane and select Filter by Current Session)

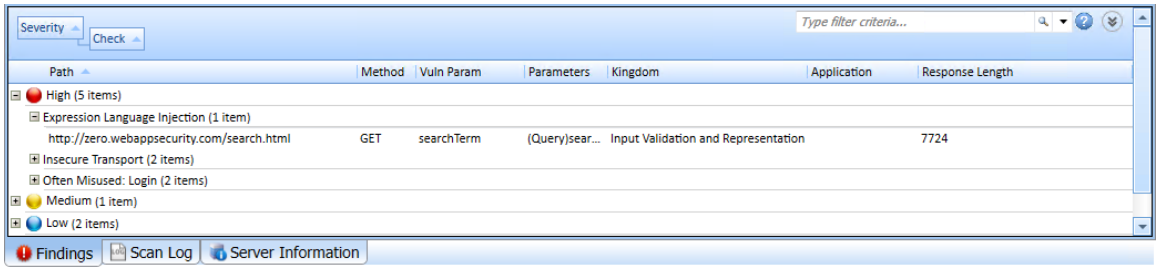
- severity - Severity assigned to a vulnerability (critical, high, medium, low)
- stack - Stack tracereturned by OpenText DAST Agent (syntax is stack:True or stack:False)
- statuscode - HTTP status code
- typerq - Type of request: query, post, or SOAP
- vparam - The vulnerability parameter

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.

The findings in the following illustration are grouped by severity and then by check name.

Summary Pane Using Groups Image



If you right-click a column header, OpenText DAST displays the following shortcut menu items related to grouping and filtering:

- 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 - Enables 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).

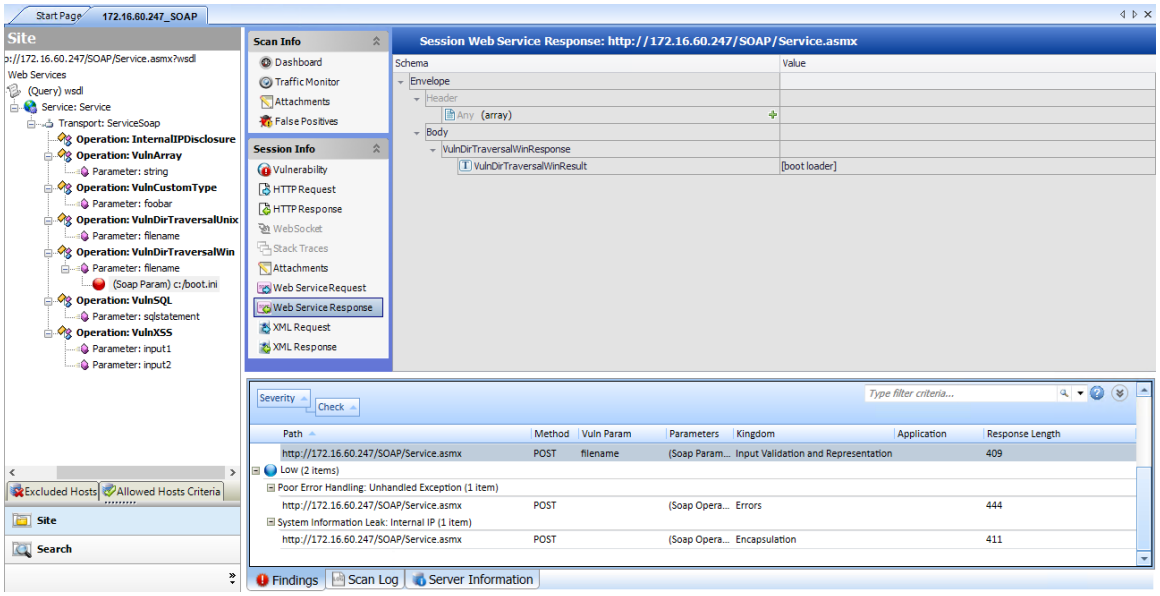
Auditing web services

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 Description Language (WSDL) 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.

Web Services Scan image



Options Available from the Session Info Panel

The following table describes the options that are available from the Session Info panel.

Option	Definition
Vulnerability	Displays the vulnerability information for the session selected in the navigation pane. For more information, see "Navigation pane" on page 46.
HTTP Request	Displays the raw HTTP request sent by OpenText DAST to the server hosting the site you are scanning.
HTTP Response	Displays the server's raw HTTP response to OpenText DAST's request. **Note:** If you select a Flash (.swf) file, OpenText DAST displays HTML instead of binary data. This allows OpenText DAST to display links in a readable format.

Option	Definition
Stack Traces	This feature is designed to support OpenText DAST Agent when it is installed and running on the target server. For certain checks (such as SQL injection, command execution, and cross-site scripting), OpenText DAST Agent intercepts OpenText DAST HTTP requests and conducts runtime analysis on the target module. If this analysis confirms that a vulnerability exists, OpenText DAST Agent appends the stack trace to the HTTP response. Developers can analyze this stack trace to investigate areas that requires remediation.
Attachments	Displays all notes, flags, and screenshots associated with the selected session. To create an attachment, do one of the following: • Right-click an operation or vulnerability in the navigation pane and select **Attachments** from the shortcut menu. • Right-click a URL on the **Findings** tab of the summary pane and select **Attachments** from the shortcut menu. For more information, see "Summary pane" on page 87. • Select an operation or vulnerability in the navigation pane, then select **Attachments** from the **Session Info** panel and click the **Add** menu (in the information pane). OpenText DAST automatically adds a note to the session information whenever you send a defect to OpenText Application Lifecycle Management (ALM).
Web Service Request	Displays an exploded view of the SOAP envelope, header, and body elements for the request.
Web Service Response	Displays an exploded view of the SOAP envelope, header, and body elements for the response.
XML Request	Displays the associated XML schema embedded in the request (available when selecting the WSDL object during a Web Service scan).
XML Response	Displays the associated XML schema embedded in the response (available when selecting the WSDL object during a Web Service scan).

For more information on how to conduct a web services vulnerability scan, see "Using the API Scan Wizard" on page 137.

Adding/viewing vulnerability screenshot

To add a vulnerability screenshot:

1. Do one of the following to select a vulnerability:
   - On the **Findings** tab in the Summary pane, right-click a vulnerable URL. For more information, see "Findings tab" on page 88.
   - On the Navigation pane, right-click a vulnerable session or URL. For more information, see "Navigation pane" on page 46.
2. On the shortcut menu, click **Attachments > Add Vulnerability Screenshot**.

   Note: An alternative method is to select a vulnerability, click **Attachments** in the **Session Info** panel, and then select a command from the **Add** menu (in the information display area). For more information, see "Information pane" on page 57.
3. If you selected a session with multiple vulnerabilities, select the check box next to one or more vulnerabilities.
4. Enter a name (40 characters max.) for the screenshot in the **Name** box.
5. Select an image file, using one of the following methods:
   - Click the browse button  and choose a file with the standard file-selection window.
   - Click **Copy from Clipboard** to save the contents of the Windows clipboard.

     Note: You can specify only one image file even if you have selected multiple vulnerabilities.
6. (Optional) Enter a note related to the vulnerability screenshot you selected.
7. Click **OK**.

Viewing screenshots for a selected session

You can view notes, flags, and screenshots for a selected session by clicking **Attachments** on the Session Info panel.

Viewing screenshots for all sessions

You can view notes, flags, and screenshots for all sessions by clicking **Attachments** on the Scan Info panel.

See also

"Vulnerability note" on page 257

"Flag session for follow-up" on page 255

"Using scan notes" on page 256

Editing vulnerabilities

After OpenText DAST 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** - OpenText DAST occasionally reports a "false positive." This occurs when OpenText DAST 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 OpenText DAST's ranking of a vulnerability, you can assign a different level, using the following scale:

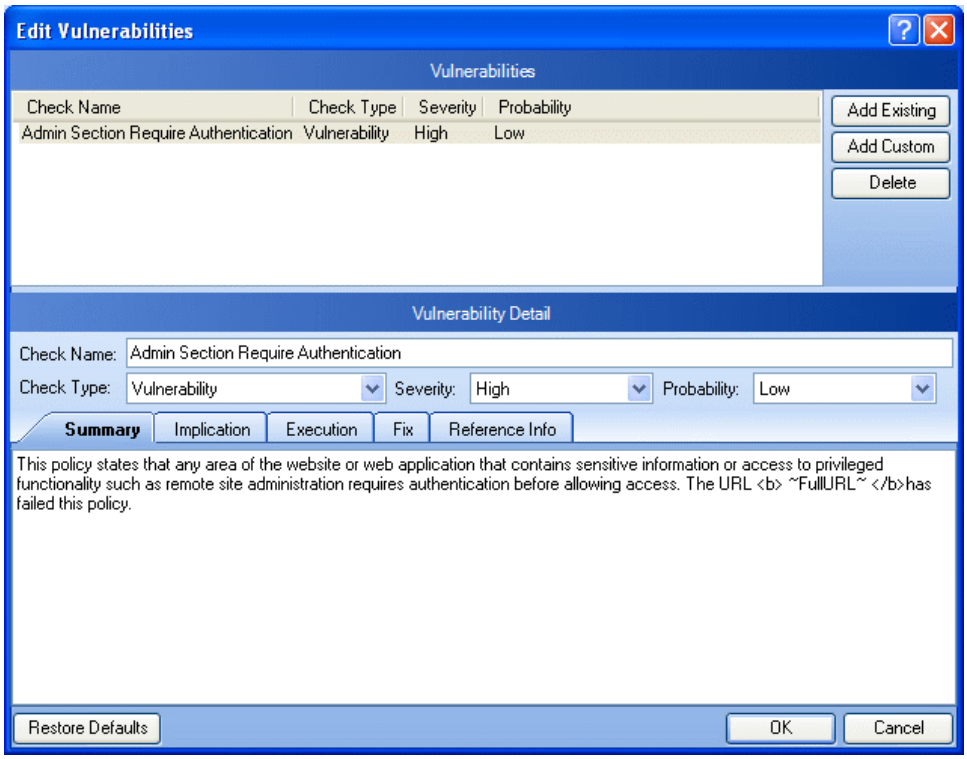
Range	Severity
0 - 9	Normal
10	Information
11 - 25	Low
26 - 50	Medium
51 - 75	High
76 - 100	Critical

- **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.

Editing a vulnerable session

To edit a vulnerable session:

1. Do one of the following to select a session:
   - On the **Findings** tab in the **Summary** pane, right-click a vulnerable URL , or
   - On the navigation pane, right-click a session or URL.
2. Select **Edit Vulnerability** from the shortcut menu.
   The Edit Vulnerabilities window opens.



3. If the session includes multiple vulnerabilities, then select a vulnerability.
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.

      Note: The * and % characters can be used interchangeably as wildcards. However, a wildcard is allowed only at the beginning, at the end, or at the beginning and end of a string. If placed within a string (such as "mic*soft,"), these characters will not function as wildcards.

   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.

Vulnerability rollup

Some sites contain a vulnerability class that is endemic throughout the site. For example, a cross-site scripting vulnerability may exist in every POST and GET method for every parameter on an entire site due to lack of input validation. This means that numerous cross-site scripting vulnerabilities will be listed on the Findings tab in the summary pane. To prevent overwhelming your development team, you can roll up such vulnerabilities into a single instance that is prefixed with the tag “[Rollup]” in OpenText DAST and reports.

What happens to rolled up vulnerabilities

When you select multiple vulnerabilities and use the rollup feature, all vulnerabilities except the first selected vulnerability are marked as ignored. The first selected vulnerability remains visible and represents the rollup. Although the rest of the selected vulnerabilities are marked as ignored, they do not appear as ignored vulnerabilities in the Recover Deleted Items window.

Caution! Rolling up vulnerabilities indicates that they share the same root cause, and that fixing the root cause will fix all rolled up vulnerabilities. Future scans will automatically ignore rolled up vulnerabilities if found. If any of the rolled up vulnerabilities do not share the same root cause, they will still be ignored.

Rollup guidelines

The following guidelines apply to vulnerability rollup:

- Scans that include vulnerability rollups can be rescanned and bulk retested.
- Only the visible vulnerability is retested during bulk retest. The rest of the vulnerabilities are ignored and will not show up as rolled up on the retest.
- Rollup is local to a scan and is not propagated between scans.
- Rollup works only when you select multiple vulnerabilities that have not been rolled up. Inadvertently selecting a currently rolled up vulnerability will prevent the Rollup

Vulnerability option from appearing in the shortcut menu.

- You can only undo a rollup if you single select a vulnerability that is currently rolled up.

Rolling up vulnerabilities

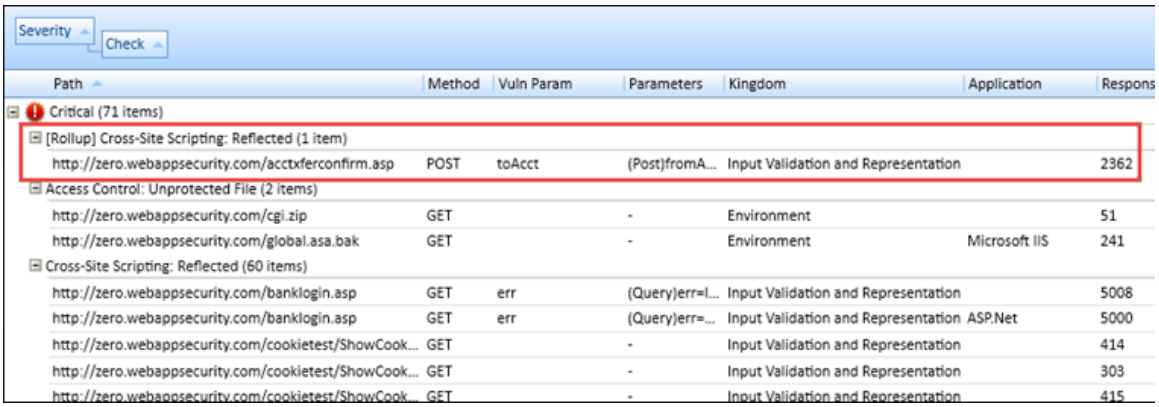
To rollup vulnerabilities:

1. On the **Findings** tab in the summary pane, select several vulnerabilities to rollup.
2. Right click and select **Rollup Vulnerabilities** from the shortcut menu.

   The following warning appears:

   Rolling up these vulnerabilities indicates that they share the same root cause, and that fixing the root cause will fix all rolled up vulnerabilities. Future scans will automatically ignore rolled up vulnerabilities if found. If any of these vulnerabilities do not share the same root cause, they will still be ignored. Do you wish to continue?
3. Do one of the following:
   - Click **OK** to rollup the vulnerabilities.
   - Click **Cancel** to leave the vulnerabilities as they are.

   If you click OK, the selected vulnerabilities are rolled into a single instance and the check name is prefixed with the tag “[Rollup]”, as shown below. Additionally, a note is added to the Attachments on the Session Info panel detailing the URLs that were rolled up and affected by the same vulnerability. For more information, see "Viewing notes for a selected session" on page 258.



Undoing rollup

The rollup feature is reversible. To undo a rollup:

1. On the **Findings** tab in the summary pane, right-click any vulnerability that has been rolled up.
2. Select **Undo Rollup Vulnerabilities**.

The rollup is reversed, and the vulnerabilities appear on the Findings tab. Additionally, the note detailing the rolled up vulnerabilities is removed from the Attachments on the Session Info panel.

Note: If you undo a rollup in a scan that has been published to Application Security, the note that was added to the Attachments on the Session Info panel detailing the roll up will be removed temporarily from OpenText DAST, but will reappear after synchronization with Application Security.

See also

"Findings tab" on page 88

Mark as false positive

If you think that OpenText DAST has erroneously determined that a session contains a vulnerability, you can use the **Mark as False Positive** dialog to remove the vulnerability from the session.

To mark as false positive:

1. Select the check box associated with one or more URLs.
2. (Optional) In the **Description** box, enter a comment or description.
3. Click **OK**.

Tip: To view a list of all sessions that have been marked as false positives, select **Suppressed Findings** from the **Scan Info** panel.

Mark as vulnerability

Use the **Mark As Vulnerability** dialog to restore the vulnerability to its original session.

1. Select the check box associated with one or more URLs.
2. (Optional) Enter a comment.
3. Click **OK**.

Flag session for follow-up

To flag a session for follow-up:

1. Do one of the following to select a session:
   - On the **Findings** tab in the Summary pane, right-click a vulnerable URL.
   - On the Navigation pane, right-click a session or URL.

2. On the shortcut menu, click **Attachments > Flag Session for Follow Up**.

   Note: You can also flag a session for follow-up by selecting a vulnerability or session, clicking **Attachments** in the Session Info panel, and then click the **Add** menu (in the information display area).

3. Enter a note related to the session you selected.
4. Click **OK**.

Viewing flags for a selected session

You can view notes, flags, and screenshots for a selected session by clicking **Attachments** on the **Session Info** panel.

Viewing flags for all sessions

You can view notes, flags, and screenshots for all sessions by clicking **Attachments** on the **Scan Info** panel.

Using scan notes

To add a scan note:

1. Click **Attachments** on the **Scan Info** panel.
2. Click **Add** and select **Scan Note**.
3. On the Add Scan Note dialog box, enter a note related to the scan.
4. Click **OK**.

To delete a scan note (or any attachment):

1. Select the attachment.
2. Click **Delete**.

See also

"Adding/viewing vulnerability screenshot" on page 250

"Vulnerability note" on the next page

"Flag session for follow-up" on the previous page

Working with session notes

To add a session note:

1. Do one of the following to select a session:
   - On the Findings tab in the Summary pane, right-click a vulnerable URL.
   - On the Navigation pane, right-click a session or URL.
2. On the shortcut menu, click **Attachments > Add Session Note**.

   Note: You can also add a session note by selecting a vulnerability or session, clicking **Attachments** in the Session Info panel, and then clicking the **Add** menu (in the information display area).
3. Enter a note related to the session you selected.
4. Click **OK**.

Viewing notes for a selected session

You can view notes, flags, and screenshots for a selected session by clicking **Attachments** on the **Session Info** panel.

Viewing notes for all sessions

You can view notes, flags, and screenshots for all sessions by clicking **Attachments** on the **Scan Info** panel.

See also

"Findings tab" on page 88

"Information pane" on page 57

"Navigation pane" on page 46

Vulnerability note

To add a vulnerability note:

1. Do one of the following to select a vulnerability:
   - On the **Findings** tab in the Summary pane, right-click a vulnerable URL. For more information, see "Findings tab" on page 88.
   - On the Navigation pane, right-click a vulnerable session or URL. For more information, see "Navigation pane" on page 46.

2. On the shortcut menu, click **Attachments > Add Vulnerability Note**.

   Note: An alternative method is to select a vulnerability, click **Attachments** in the Session Info panel, and then click the **Add** menu (in the information display area). For more information, see "Information pane" on page 57.

3. If you selected a session with multiple vulnerabilities, select the check box next to one or more vulnerabilities.
4. Enter a note related to the vulnerability (or vulnerabilities) you selected.
5. Click **OK**.

Viewing notes for a selected session

You can view notes, flags, and screenshots for a selected session by clicking **Attachments** on the **Session Info** panel. If the selected session includes rolled up vulnerabilities, a note in the Description area details the URLs that were rolled up and affected by the same vulnerability. For more information, see "Vulnerability rollup" on page 253.

Viewing notes for all sessions

You can view notes, flags, and screenshots for all sessions by clicking **Attachments** on the **Scan Info** panel.

Recovering deleted sessions

When you remove a session, OpenText DAST deletes the session from the Navigation pane (in both the Site and Sequence views) and from the **Findings** tab in the Summary pane. It also omits the session from any reports you may generate.

The number of deleted sessions is displayed on the Dashboard (under the Scan category). To recover removed sessions:

1. Click the highlighted number that appears next to the **Deleted Items** header.

   The Recover Deleted Items window displays a list of deleted items.
2. Select the check box next to each session that you want to recover.
3. To view detailed information about the session, select **Show details when selected**.
4. Click **Recover** and then click **Yes** when prompted to verify your selection.

Recovered sessions reappear along with their parent sessions in the Navigation pane (both Site and Sequence views) and in the **Findings** tab in the Summary pane.

See also

"Session Info panel" on page 69

Sending vulnerabilities to OpenText ALM

You can convert one or more vulnerabilities to defects and add them to the OpenText Application Lifecycle Management (ALM) database.

To send a vulnerability to your defect tracking system:

1. Right-click a vulnerability in either the Navigation pane or the Summary pane. For more information, see "Navigation pane" on page 46 and "Summary pane" on page 87.
2. Select **Send to** and choose **OpenText ALM**.
3. On the Send to dialog box, choose a profile from the **Profile** list.

   If you need to create or edit a profile, click **Manage** to access the OpenText DAST Application Settings. For more information, see "Application settings: OpenText ALM" on page 456.

   Note: If the selected profile maps an OpenText DAST vulnerability to "Do not publish" (based on its severity level), the vulnerability will not be exported.
4. To force the creation of a defect even if it has been previously reported, select **Allow duplicate defect assignment**.

   OpenText DAST recognizes duplicates only within the same scan. If you scan a site and send a specific vulnerability to ALM, you can prevent OpenText DAST from sending that same vulnerability if it is encountered again during that scan. However, if you conduct a subsequent scan of that site and OpenText DAST again encounters that same vulnerability, OpenText DAST is not programmatically aware that the vulnerability was sent to ALM during the previous scan.
5. To close this dialog box after sending the defect(s), select **Close when finished**.
6. If you have selected multiple vulnerabilities, you can exclude a vulnerability by removing the check mark next to the ID number.
7. Click **Send**.

Additional information sent

OpenText DAST will also add a note to the session information indicating that the defect was sent to OpenText ALM, as illustrated by the following example:

Defect #30 was created in OpenText ALM.
Check ID: 182
CheckName: Dan-o Log Information Disclosure
Profile: QA-user1
Server URL: http://myvm2023/qcbin
Project: test3
Priority: 3-High
Severity: 1-Low

Note: If you receive the error message, "Error authenticating with OpenText ALM," see "Disabling Data Execution Prevention " below.

Disabling Data Execution Prevention

When you attempt to integrate with OpenText Application Lifecycle Management (ALM), you may receive the error message:

```
Error authenticating with OpenText ALM.
```

If so, you must disable Microsoft's Data Execution Prevention (DEP). For instructions on changing DEP settings, refer to your Windows documentation.

Generating a report

You can launch the Report Generator using a variety of methods:

- On the Start page, click **Generate a Report** in the left pane of the client area.
- On the OpenText DAST toolbar, click **Reports**.
- Click the **Reports** menu and select **Generate Report**.
- On the Manage Scans form, right-click a scan name and select **Generate Report**.
- With a scan open, right-click a session in the Site view and select **Generate Session Report**. For more information, see "Site view" on page 48.
- When scheduling scans.

To generate a report:

1. Launch the Report Generator using one of the options listed above.
2. Select one or more scans from the Select a Scan window.
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.
4. Click **Next**.
5. (Optional) Select a report from the **Favorites** list.

   Tip: "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. See "Standard reports" on page 263 for report descriptions.
7. Provide information for any parameters that may be requested. An exclamation mark indicates a required parameter.
8. If you want to display each report on a separate tab (rather than combining all reports

on one tab), select **Open Reports in Separate Tabs**.

9. Click **Finish**.

Saving a report

After OpenText DAST generates and displays the report, you can save it 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 OpenText DAST internal format (.raw)
- Rich Text Format (.rtf)
- Text (.txt)
- Microsoft Excel (.xls)

See also

"Standard reports" on page 263

"Advanced report options" below

"Compliance templates " on page 265

"Application settings: Reports" on page 452

Advanced report options

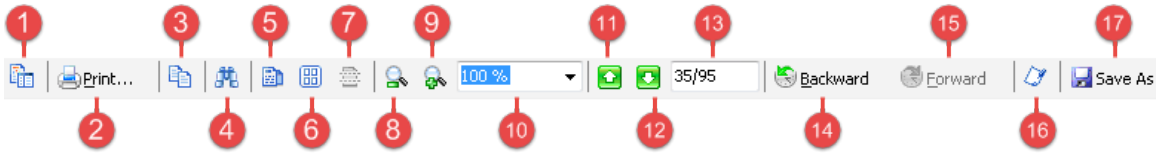
The following table describes the advanced report options:

Option	Description
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>*

Option	Description
	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.
Header/Footer Report	Select a format for the report's header and footer, and then enter or select the components.

Report viewer

Use the toolbar to navigate through the report, print or save the report, and to add notes.



Item	Description
1	Show / Hide Table of Contents
2	Print Report
3	Copy
4	Search
5	Single Page View
6	Multi-Page View
7	Continuous Scroll
8	Zoom Out
9	Zoom In
10	Magnification

Item	Description
11	Previous Page
12	Next Page
13	Current Page Number / Total Number of Pages
14	Page Backward
15	Page Forward
16	Annotation (see "Adding a note" below)
17	Save Report

Note: The Backward and Forward buttons function in the same manner as the Back and Forward buttons on a browser. They navigate forward or backward one step in the history list.

Adding a note

To add a note:

1. Click the Annotation icon.
2. Select a format.
3. Drag it to the report.
4. Right-click the note and select **Properties**.
5. Select the Text property and enter the contents of the note.

Standard reports

The following table describes the standard reports that are available.

Report	Description
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

Report	Description
	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.
Crawled URLs	For each URL encountered during the crawl, this report lists any cookies sent and the raw HTTP request and response.
Developer Reference	Totals and detailed description of each form, JavaScript, email, comment, hidden control, and cookie discovered on the website. You can select one or more of these reference types.
Duplicates	This report contains information about vulnerabilities detected by OpenText DAST Agent 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 OpenText DAST 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

Report	Description
	occur in one website but not the other.
Scan Log	Sequential list of the activities conducted by OpenText DAST during the scan (as the information appears on the **Scan Log** tab of the summary pane).
Trend	This report enables 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.

Manage reports

Use Manage Reports to rename, add, delete, or import report definition files.

Note that standard reports cannot be renamed, deleted, or exported.

Compliance templates

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

Note: This list might not match the templates that you see in your product. SmartUpdate might have added templates since this document was produced.

Template	Description
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

Template	Description
	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/ICE 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

Template	Description
	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.
CWE Top 25 ***<version>***	The Common Weakness Enumeration (CWE) Top 25 Most Dangerous Software Errors (CWE Top 25) is a list of weaknesses created by MITRE that demonstrates the most widespread and critical weaknesses that can lead to serious vulnerabilities in software. MITRE outlines its methodology as follows: "To create the list, the CWE Team used a data-driven approach that leverages published Common Vulnerabilities and Exposures (CVE) data and related CWE mappings found within the National Institute of Standards and Technology (NIST) National Vulnerability Database (NVD), as well as the Common Vulnerability Scoring System (CVSS) scores associated with

Template	Description
	each of the CVEs. A scoring formula was then applied to determine the level of prevalence and danger each weakness presents. This data-driven approach can be used as a repeatable, scripted process to generate a CWE Top 25 list on a regular basis with minimal effort."
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 life cycle 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	This compliance template reports all applicable web application components of the Application Security and Development

Template	Description
STIG *<version>*	Security Technical Implementation Guide (STIG) Version 3, Release 2. 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.
DoD Control Correlation Identifier (CCI)	The Defense Information Systems Agency (DISA) Field Security Operations (FSO) created the CCI specification and is currently responsible for the maintenance of the CCI specification and CCI List. The Control Correlation Identifier (CCI) provides a standard identifier and description for each of the singular, actionable statements that comprise an Information Assurance (IA) control or IA best practice. CCI bridges the gap between high-level policy expressions and low-level technical implementations. CCI allows a security requirement that is expressed in a high-level policy framework to be decomposed and explicitly associated with the low-level security setting(s) that must be assessed to determine compliance with the objectives of that specific security control. This ability to trace security requirements from their origin (such as regulations, IA frameworks, and so forth) to their low-level implementation allows organizations to readily demonstrate compliance to multiple IA compliance frameworks. CCI also provides a means to objectively rollup and compare related compliance assessment results across disparate technologies. This report maps the OpenText Fortify 7PK Taxonomy to DISA CCI.
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

Template	Description
	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.
General Data Protection Regulation (GDPR)	The EU General Data Protection Regulation (GDPR) replaces the Data Protection Directive 95/46/EC and was designed to harmonize data privacy laws across Europe, to protect and empower all EU citizens' data privacy, and to reshape the way organizations across the region approach data privacy. In effect as of May 25, 2018, GDPR provides a framework for

Template	Description
	organizations on how to handle personal data. According to GDPR regulation, personal data "means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier, or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural, or social identity of that natural person." GDPR articles that pertain to application security and require businesses to protect personal data during design and development of its product and services are: • Article 25, Data protection by design and by default - which requires implementation of "appropriate technical and organizational measures for ensuring that, by default, only personal data which are necessary for each specific purpose of the processing are processed." • Article 32, Security of processing - which requires businesses to protect its systems and applications from "from accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to personal data." This report may be used by organizations as a framework to help identify and protect personal data as it relates to application security.
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 compliance template as a baseline in crafting a compliance policy to meet the needs of your organization and its security policy.

Template	Description
ISO27001 *<version>*	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 *<version>*	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

Template	Description
	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 ASVS	The Open Web Application Security Project (OWASP) Application Security Verification Standard (ASVS) is a list of application security requirements or tests that can be used by architects, developers, testers, security professionals, tool vendors, and consumers to define, build, test, and verify secure applications **Note:** Some mapping to the CWE category in the OWASP ASVS document does not match the intent of the category or matched in a limited scope. Review the reported CWE mappings in reports generated with this compliance template.
OWASP Top Ten *<year>*	Many government agencies suggest testing for the OWASP Top Ten Web application vulnerabilities as a best practice in ensuring the security of your Web application.
PCI Data Security *<version>*	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.
PCI SSF *<version>*	This compliance template applies to the application security portions of the Secure Software Requirements and Assessment Procedures defined in the Payment Card Industry (PCI) Software Security Framework (SSF). WebInspect tests for 18

Template	Description
	application security-related control objectives across Control Objective sections 2, 3, 4, 5, 6, 7, 10, A.2, C.1, C.2, C.3, and C.4 of PCI SSF and reports whether each control objective is In Place or Not In Place to indicate whether requirements are satisfied or not. This report is intended to measure the level of adherence the specific application(s) possess when compared to PCI SSF compliance and is not intended to serve as a comprehensive Report on Compliance (ROC). The information contained in this report is targeted at project managers, security auditors, and compliance auditors.
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

Template	Description
	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 *<version>*	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. **Note:** SANS compliance templates other than "CWE" may also be available.
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.
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:

Template	Description
	• 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 *<version>*	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.

Managing settings

The Manage Settings window enables 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** and select **Default Scan Settings**.

Accessing the Manage Settings window

To access the Manage Settings window:

- From the **Edit** menu, select **Manage Settings**.
  The Manage Settings window opens.

Creating a Settings File

To create a settings file:

1. Click **Add**.
2. On the Create New Settings window, change settings.
3. When finished, click **OK**.
4. Using a standard file-selection dialog box, name and save the file.

Editing a Settings File

To edit a settings file:

1. Select a file.
2. Click **Edit**.
3. On the Create New Settings window, change settings.
4. When finished, click **OK**.

Deleting a Settings File

To delete a settings file:

1. Select a file.
2. Click **Delete**.

Importing a Settings File

To import a settings file:

1. Click **Import**.
2. Using a standard file-selection dialog box, select a settings file and click **Open**.

Exporting a Settings File

To export a settings file:

1. Select a file.
2. Click **Export**.
3. Using a standard file-selection dialog box, name the file and select a location.
4. Click **Save**.

Scanning with a Saved Settings File

To scan with a saved settings file:

1. From the **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 box, select the settings file you want to use and click **Open**.

The file you select is now your default settings file.

SmartUpdate

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

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

You can also run SmartUpdate on demand through the OpenText DAST user interface by selecting **Start SmartUpdate** from the OpenText DAST **Start Page**, by selecting **SmartUpdate** from the Tools menu, or by clicking the **SmartUpdate** button on the standard toolbar. For more information, see "Tools menu " on page 41 and "Toolbars" on page 43.

For installations lacking an Internet connection, see "Performing an offline SmartUpdate" on page 281.

Performing a SmartUpdate (Internet connected)

To perform a SmartUpdate when OpenText DAST is connected to the Internet:

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

   If updates are available, the SmartUpdater window opens with the Summary tab in view. The Summary tab displays up to three separate collapsible panes for downloading the following:
   - New and updated checks
   - OpenText DAST software
   - SmartUpdate software
2. Select the check box associated with one or more of the download options.
3. (Optional) To view details about the checks being updated:
   a. Click the **Check Detail** tab.

      In the left pane is a list showing the ID, Name, and Version of checks being updated. The list is grouped by Added, Updated, and Deleted.
   b. To view the policies that include a specific check being updated, select the check in the list.

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

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

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

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

Downloading checks without updating OpenText DAST

Engine updates are required for some checks to be run during scans. If you are not using the latest version of OpenText DAST, it is likely that some of the checks in your

SecureBase cannot be run during a scan. To test your application with all the latest checks, ensure that you are using the latest version of OpenText DAST.

Performing an offline SmartUpdate

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

Stage	Description
1.	Open a support case. Customer Support personnel will provide you with the offline FTP server URL and login credentials (if needed). For more information, see "Preface" on page 24.
2.	On a machine that can access the Internet, access the offline FTP server.
3.	Download the OpenText DAST static SmartUpdate ZIP file.
4.	On the machine where OpenText DAST is installed, extract all files from the ZIP file.
5.	Close OpenText DAST.
6.	Copy the extracted **SecureBase.db** and **version.txt** files to the directory where your SecureBase data resides. The default location is: `C:\ProgramData\HP\HP WebInspect\SecureBase` **Tip:** By default, these folders are hidden in Windows. Be sure to change folder options to show hidden files.

WebSphere Portal FAQ

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_
B3xmpzBfnacLv6QuIfxVHKJGhmNfzToue8nWdKg4fx8jtaT9MJpB2zQPgqLp9GrADyey0tvvL1F9Sn
ftm_
y0cbuw8Xbmvg2NN6412wlsQP27GAa3AO9AEBJhmxxcnWHlk8kverBIBQ!!/dl4/d5/L2dBISEvZ0FB
IS9nQSEh/
```

Which versions of WebSphere Portal are supported?

Versions 6.1 and later are supported.

Why does OpenText DAST 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, OpenText DAST 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 OpenText DAST decode the navigation state?

WebSphere Portal 6.1 and later include a URL decoding service. When the WebSphere Portal overlay is selected, OpenText DAST 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 OpenText DAST, 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?

OpenText DAST'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 OpenText DAST can perform on WebSphere Portal applications?

OpenText DAST can perform all manipulation attacks on WebSphere Portal applications. This includes (but is not limited to) XSS, SQL Injection, CSRF, RFI, LFI and others. OpenText DAST 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 Customer 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 Customer Support for assistance.

For more information, see "Preface" on page 24.

Command-line execution

OpenText DAST includes the following applications that you can use by way of the command-line interface (CLI):

- WI.exe – enables you to configure and conduct a scan using an existing macro, export scan files and reports, merge scans, reuse scans, and test the login macro of an existing scan. For more information, see "Using wi.exe" on the next page.
- WIScanStopper.exe – enables you to stop a scan that is currently running. For more information, see "Using WIScanStopper.exe" on page 303.
- MacroGenServer.exe – enables you to create a login macro. For more information, see "Using MacroGenServer.exe" on page 303.

These applications are installed in the same directory as OpenText DAST. By default, the installation directory is:

```
C:\Program Files\Fortify\Fortify WebInspect
```

Launching the CLI

To launch the CLI:

- Right-click the Windows **Command Prompt** (`cmd.exe`) application, and select **Run as administrator**.

The Administrator: Command Prompt window appears.

Important! At the command prompt, use the `cd` command to change the current working directory to the directory where the applications are installed.

CLI limitations in OpenText DAST on Docker

Some parameters and features accessible from the command-line interface are not supported in OpenText DAST on Docker. Items that are not supported are indicated as such.

Using wi.exe

You can initiate several OpenText DAST functions by way of a command-line interface (CLI) using the program `wi.exe`. Use the following syntax when typing a command:

```
wi.exe -u url [-api type] [-s file] [-ws file] [-Framework name]
       [-CrawlCoverage name] [-ps policyID | -pc path]
       [-ab|ac|an|ad|aa|ak|at creds] [-macro path] [-o|c] [-n name]
       [-e[abcdefghijklmnopst] file] [-x|xd|xa|xn] [-b filepath] [-db]
       [-d filepath -m filename] [-i[erxd] scanid | -ic scanid scanname
       | -im option scanid scanlist] [-r report_name -y report_type
       -w report_favorite -f report_export_file -g[phacxe]
       [-t compliance_template_file] [-v] [-?]
```

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\Fortify\Fortify 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
```

Options

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

Category	Options	Definition
General	`-?`	Displays the usage help.
	`-u {url}`	Specifies the start URL or IP address.

Category	Options	Definition
		Caution! 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. If the URL contains an ampersand (&), you must enclose the URL within quotation marks.
	`-api {type}`	Specifies the API type to be scanned. Valid values for *type* are: `GraphQL` `gRPC` `OData` `SOAP` `Swagger` **Important!** You must provide the URL to the Swagger or OData definition file, as shown in the following example: `-u http://172.16.81.36/v1 -api Swagger` **Important!** For the `-u` option, you can point to a definition file or an endpoint for the service, as shown in the following example: `-u http://172.16.81.36/v1 -api Swagger` Optionally, you can create a scan configuration file with

Category	Options	Definition
		additional information, such as authentication and proxy settings, and point to the settings file in the command. For more information, see "Scanning an API with wi.exe" on page 155.
	`-s {filename}`	Specifies the settings file. Settings file types are JSON and XML. **Note:** Command line parameters take precedence over values in a settings file.
	`-db`	Indicates to use the database defined in settings file. If omitted, OpenText DAST defaults to database connection defined in application settings.
	`-ws {filename}`	Identifies the web service design file to use.
	`-o`	Specifies an Audit-only scan.
	`-c`	Specifies a Crawl-only scan.
	`-n {name}`	Specifies the scan name.
	`-b {filepath}`	Specifies the SecureBase file to use. For path, specify the full path and file name.
	`-d {filepath}`	Moves the database to the specified filepath.
	`-m {filename}`	Moves the database to specified filename.
	`-v`	Creates verbose output.

Category	Options	Definition
	`-tm`	Enables the Traffic Monitor (Traffic Viewer) for the scan. **Important!** The Traffic Monitor requires the traffic session file (`.tsf`) from the scan. If a scan with Traffic Monitor enabled needs to be exported to a scan file in the `.scan` format, use the `-et` option to export all scan logs, including the traffic session file.
	`-ie {scanid}`	Starts configured scan with the specified scan ID (GUID).
	`-ir {scanid}`	Resumes scan with the specified scan ID (GUID).
	`-ix {scanid}`	Uses existing scan with the specified scan ID (GUID), but does not continue the scan.
	`-id {scanid}`	Deletes scan with the specified scan ID (GUID).
	`-ii {scanid} {file path}`	Imports scan. **Note:** This parameter is not supported in OpenText DAST on Docker.
Restrict to Root Folder	`-x`	Restricts scan to directory only (self).
	`-xa`	Restricts scan to directory and parents (ancestors).
	`-xd`	Restricts scan to directory and subdirectories (descendants).
	`-xn`	Ignores “restrict to folder” rules in referenced settings file.

Category	Options	Definition
		Restrict to folder parameters (`x|xa|xb|xn`) can be in their own category (as report or output).
Framework	`-framework {framework_name}`	Specifies 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}`	Specifies the type of scan coverage. 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 {policy id}` **Tip:** You can specify multiple policies as a comma-separated list of policy IDs. For example: `-ps 1001,1002` If you specify multiple policies, the sensor will aggregate the policies during the scan.	Identifies the non-custom policy to use. Values for *policy id* are as follows: **Best Practices** `1` = Standard `1012` = OWASP Top 10 Application Security Risks 2013 `1024` = SANS Top 25 2011 `1025` = OWASP Top 10 2017 `1027` = General Data Protection Regulation (GDPR) `1034` = DISA-STIGV4R9 `1036` = DISA-STIGV4R10 `1037` = CWE Top 25 `1041` = OWASP Application Security Verification Standard

Category	Options	Definition
		(ASVS) 1043 = DISA-STIGV4R11 1044 = API 1045 = DISA-STIGV5R1 1046 = NIST-SP80053R5 1047 = CWE Top 25 2020 1048 = CWE Top 25 2021 1049 = OWASP Top 10 2021 **By Type** 3 = SOAP 7 = Blank 1001 = SQL Injection 1002 = Cross-Site Scripting 1005 = Passive 1008 = Critical and High Vulnerabilities 1010 = Aggressive SQL Injection 1011 = NoSQL and Node.js 1013 = Mobile 1015 = Apache Struts 1016 = Transport Layer Security 1020 = Privilege Escalation 1021 = Server-side 1022 = Client-side 1026 = DISA-STIG-V4R4 1029 = DISA-STIG-V4R5 1030 = DISA-STIG-V4R6 1031 = DISA-STIG-V4R7 1032 = DISA-STIGV4R8 1033 = WebSocket 1035 = PCI Software Security Framework 1.0 (PCI SSF 1.0) 1050 = OAST 1055 = PCI DSS 4.0 1056 = OWASP API Top 10 - 2023 **Deprecated** 2 = Assault (Deprecated) 4 = Quick (Deprecated) 5 = Safe (Deprecated)

Category	Options	Definition
		`6` = Development (Deprecated) `16` = QA (Deprecated) `17` = Application (Deprecated) `18` = Platform (Deprecated) `1009` = OWASP Top 10 Application Security Risks 2010 (Deprecated) `1014` = OpenSSL Heartbleed (Deprecated) `1018` = Standard (Deprecated) `1019` = Deprecated Checks `1051` = Aggressive Log4Shell (Deprecated) **Hazardous** `1004` = All Checks
	`-pc {policy path}`	Specifies a custom policy to use. For path, specify the full path and file name, such as: `C:\ProgramData\hp\HP WebInspect\MyCustomPolicy.policy`
Authentication	`-ab "userid:pwd"`	Specifies Basic mode (user name and password).
	`-ac "userid:pwd"`	Specifies ADFS CBT mode (user name and password).
	`-an "userid:pwd"`	Specifies NTLM mode (user name and password).
	`-ad "userid:pwd"`	Specifies Digest mode (user name and password).
	`-aa "userid:pwd"`	Specifies Automatic mode (user name and password).
	`-ak "userid:pwd"`	Specifies Kerberos mode (user name and password).
	`-am {macro path}`	Deprecated; use the `-macro`

Category	Options	Definition
		option.
	`-at "{type} {token}"`	Specifies the authentication mode (type and token) for API scans, such as: `-at "Basic YWxh0GRpbjpvcGVuc2VzYW1l"` Authentication modes for *type* are as follows: `Basic` `Bearer` `Digest` `HOBA` `Mutual` `Negotiate` `OAuth` `SCRAM-SHA-1` `SCRAM-SHA-256` `vapid` **Note:** The type and token must be enclosed in double quotation marks as shown previously.
Macro	`-macro {macro path}`	Specifies macro name and directory path for web macro authentication.
	`-macro {url} {username} {password}`	Creates auto-generated macro for authentication.
Login Macro Parameters	`-ls "userid:pwd"`	Replaces the SmartCredentials UserName and Password with the supplied values.
	`-lt "name0:value0;name1:value1;...nameN:valueN"`	Replaces existing TruClient login parameters that match the specified names.
Output	`-ea {filepath}`	Exports scan in legacy full XML format.

Category	Options	Definition
	`-eb {filepath}`	Exports scan details (Full) in legacy XML format.
	`-ec {filepath}`	Exports scan details (Comments) in legacy XML format.
	`-ed {filepath}`	Exports scan details (Hidden Fields) in legacy XML format.
	`-ee {filepath}`	Exports scan details (Script) in legacy XML format.
	`-ef {filepath}`	Exports scan details (Set Cookies) in legacy XML format.
	`-eg {filepath}`	Exports scan details (Web Forms) in legacy XML format.
	`-eh {filepath}`	Exports scan details (URLs) in legacy XML format.
	`-ei {filepath}`	Exports scan details (Requests) in legacy XML format.
	`-ej {filepath}`	Exports scan details (Sessions) in legacy XML format.
	`-ek {filepath}`	Exports scan details (E-mails) in legacy XML format.
	`-el {filepath}`	Exports scan details (Parameters) in legacy XML format.
	`-em {folderpath}`	Exports scan details (Web Dump) in legacy XML format.
	`-en {filepath}`	Exports scan details (Offsite Links) in legacy XML format.
	`-eo {filepath}`	Exports scan details (Vulnerabilities) in legacy XML format.
	`-ep {filepath}`	Exports scan in FPR format to specified file.

Category	Options	Definition
	`-eq {format} {filepath}`	Exports scan details from the Site Tree. The details include: • Date and time (in milliseconds) the request was sent • Host • Path • Method • Status code • Elapsed time (in milliseconds) between the request and the response For single-page application (SPA) scans: • The CSV format includes SPADisplayName and SPASelector columns. • The JSON format includes SPA Events which contain SPADisplayName and SPASelector data. For more information, see "SPA coverage" on page 51. Values for format are: `json` `csv` If a value being exported includes double quotation marks, escape characters (double quotation marks) will be added to the CSV output. For example, the selector "Sign in" includes double quotation marks, so it will appear as follows in the CSV file: `"//a[normalize-space(string`

Category	Options	Definition
		`(.))=""Sign in""]"` **Tip:** Use this option in conjunction with the `-ie`, `-ir`, `-ix`, or any of the start scan options to identify the scan for which you want to retrieve data. For example: `-ix {scan GUID} -eq {format} {filepath}`
	`-es {filepath}`	Exports scan in `.scan` format to specified file.
	`-et {filepath}`	Exports scan with logs in `.scan` format to specified file.
	`-eu {filepath}`	Exports scan settings to specified file after applying all other overrides. **Note:** This parameter does not run the scan. It exports the settings and exits.
Reports	`-r {report_name}` **Tip:** For multiple reports, separate report names with a semicolon. All reports will be contained in a single file.	Identifies the name of the report to run. Valid values for *report_name* are: `Aggregate` `Alert View` `Attack Status` `Compliance` `Crawled URLs` `Developer Reference` `Duplicates` `Executive Summary` `False Positive` `QA Summary` `Scan Difference` `Scan Log` `Trend` `Vulnerability`

Category	Options	Definition
		`Vulnerability (Legacy)` **Note:** Report names containing a space must be enclosed in quotation marks.
	`-w {favorite_name}`	Identifies the name of the report favorite to run.
	`-ag`	Aggregates reports in report favorite.
	`-y {report_type}`	Specifies the type of report: `Standard` or `Custom`.
	`-f {export_file}`	Specifies the file path and file name where the report will be saved.
	`-gp`	Exports as Portable Document Format (PDF) file.
	`-gh`	Exports as HTML file.
	`-ga`	Exports as raw report file.
	`-gc`	Exports as rich text format (RTF) file.
	`-gx`	Exports as text file.
	`-ge`	Exports as Excel file.
	`-t {filepath}`	Specifies compliance template file to use.
Scan Merge	`-ic {scan id} {scan name}`	Creates a merge target scan. For more information, see "Merging scans" on page 302 in this topic. **Note:** This parameter is not supported in OpenText DAST on Docker.

Category	Options	Definition
	`-im /o:{option} {merge target scan id} {source scan id1} {source scan id2}`	Merges scans. For more information, see "Merging scans" on page 302 in this topic. Choices for *option* are: • `Replace` - Replace target session and vulnerabilities with source session and vulnerabilities. • `ReplaceMergeVulns` - Replace target session with source session, and add source vulnerabilities to target scan. • `Skip` - When session IDs are the same in both scans, do not merge sessions or vulnerabilities. • `SkipMergeVulns` - When session IDs are the same in both scans, do not replace target session and copy vulnerabilities from source. • `Smart` - Consider source and target policy and times when merging. **Important!** Use the `-ic` parameter to create the merge target scan before using the `-im` parameter. **Note:** This parameter is not supported in OpenText DAST on Docker.
Scan Reuse	`-iz /o:{option} {source scan id} {settings filename}`	Creates reuse scan settings. Choices for *option* are: • `Incremental` - Use same settings as source scan, with a

Category	Options	Definition
		modified policy that disables checks that flagged in source scan and that should only flag once. This mode audits only new crawl surface. A new crawl is performed, but only new sessions are audited. • `Remediation` - Use same settings as source scan, with a modified policy that disables checks that did not flag in source scan. The *settings filename* is the name of the modified settings file being created. **Note:** This parameter is not supported in OpenText DAST on Docker.
Scan Findings Retest	`-iv <guid> {[<severity> | <vuln ID prefix>] ...} /s <file path>`	Creates a settings file that you can use to start a scan to retest findings. You can retest findings by severity or unique sessionCheckFoundID or both. If you do not provide a severity or sessionCheckFoundID, then all findings in the base scan are retested. Parameter components are as follows: • *<guid>* is the base scan ID. This is required. • *<severity>* is the vulnerability severity or severities to retest. All vulnerabilities from the base scan that were flagged with the listed severity or severities will be retested. Options for severity are: `Critical`, `High`, `Medium`,

Category	Options	Definition
		`Low`. • *`<vuln ID prefix>`* is the unique sessionCheckFoundID, which can be retrieved by way of the SessionCheckFounds API endpoint. For more information, see the OpenText DAST REST API Swagger UI. **Tip:** You can specify a prefix of the sessionCheckFoundID. For example, `012f` would match sessionCheckFoundID `012fa34124`. • */s <file path>* is the directory path and file name for the vulnerability retest settings file that will be created. This parameter is required, and modifies the settings from the original scan to specify a retest. The new settings file that is created identifies the vulnerability or vulnerabilities being retested. You can provide a list consisting of severities and sessionCheckFoundIDs in any order. The following example shows a valid list: `Critical 3156 High 1234`
Test Login Macro	`-it {scan id}`	Tests login macro of existing scan.
Selenium Macro	`-selenium_workflow {ArrayOfSelenium Command`	Creates a Selenium workflow scan.

Category	Options	Definition
	`object}`	For the complete process and procedures involved in using this command, see "Integrating with Selenium WebDriver" on page 328.
	`-selenium_no_ validation`	Disables validation of Selenium commands before running the scan. **Important!** When using this parameter, you must specify one or more allowed hosts. For more information, see "Integrating with Selenium WebDriver" on page 328.
	`-slm {SeleniumCommand object}` or `@"PathtoFilewithobject"`	Specifies a Selenium login macro for the scan. This option uses the `ArrayOfSeleniumCommand object` with one element or the `SeleniumCommand object`. Use `@"PathtoFilewithobject"` to specify the path to a file that includes the `SeleniumCommand object` or `ArrayOfSeleniumCommand object`. See "Selenium login macro example" on page 301. **Important!** A `LogoutCondition` element is required.
Postman Scans	`-pwc {filename}`	Starts a scan with a Postman Collection file. This option can accept several collection files separated by commas, such as: `-pwc pcOne,pcTwo,pcThree` For more information, see

Category	Options	Definition
		"Scanning with a Postman collection" on page 319.
	`-pdac`	Disables Postman auto-configuration so that auto-configuration or analysis of the Postman collection is not performed before the scan.
	`-plc {Collection path}`	Specifies the path to the Postman login collection.
	`-pls "logoutsignature"`	Identifies the logout condition. This parameter accepts Regex Extensions. **Important!** You must replace the space character with `\s`.
	`-pec {filepath}`	Specifies the Postman environment file to be used in the scan.
	`-pg {filepath}`	Specifies the Postman global variables file to be used in the scan.
State Management	`-rs {<ArrayOfResponse StateElement>} or "@{file path}"`	Supplies a response state rule. This parameter accepts an `ArrayOfResponseStateElement` element or a response state rule stored in a file. It is used for Bearer token and API Key. **Important!** To use a response state rule stored in a file, you must specify the file path with the `@` symbol. For examples, see "Response state rule example" on the next page.

Category	Options	Definition
Other Settings	`-ah {url} [,{url},...]`	Lists the Allowed Hosts. The `URL` is the schema, host, and port number.

Examples

The following examples illustrate command line execution as if executed from the OpenText DAST home directory:

```
wi.exe -u www.anywebsite.com -ps 1 -ab MyUsername:Mypassword
```

```
wi.exe -u https://zero.webappsecurity.com
       -s c:\program files\webinspect\scans\scripted\
       -r "Executive Summary";Vulnerability -y Standard
       -f c:\program files\webinspect\scans\scripted\zero051105.xml -gx
```

If you do not specify a policy, OpenText DAST will crawl (but not audit) the website.

If you specify an invalid policy number, OpenText DAST will not conduct the scan.

Selenium login macro example

The following is an example of the Selenium login macro option:

```
-slm "<SeleniumCommand><Command>"wi command\"</Command>
   <AllowedHosts><string>http://hostname/</string>
   </AllowedHosts><LogoutCondition>Access\sDenied</LogoutCondition>
   </SeleniumCommand>"
```

Response state rule example

The following is an example of a response state rule:

```
-rs "<ArrayOfResponseStateElement><ResponseStateElement><name>
   AutoDetect</name><ReplaceRegexes><string>Authorization:\sBearer\s
   (?&lt;AutoDetect&gt;[^\r\n]*)\r\n</string></ReplaceRegexes>
   <SearchRegexes><string>""en"":""(?&lt;AutoDetect&gt;
   [-a-zA-Z0-9._~+/]+?=*)""}$</string></SearchRegexes>
   </ResponseStateElement></ArrayOfResponseStateElement>"
```

Tip: You can create response state rules in Scan Settings: HTTP Parsing in the OpenText DAST user interface. You can then open the scan settings XML file, locate the `ResponseStateElement`, and copy and paste it into the `-rs` parameter. For more information about response state rules, see "Scan settings: HTTP Parsing" on

page 376.

The following code shows an example starting a Postman scan using a response state rule that is stored in a file:

```
wi -pwc c:\BearerWorkflow.json -pdac -plc c:\BearerLogin.json
   -rs @c:\BearerResponseStateRule.txt -pls
```

Merging scans

Note: This feature is not supported in OpenText DAST on Docker.

You cannot merge into an existing scan. You must first create a merge target using the "ic" parameter.

The scans to be merged are sorted by scan date and are merged in that order. Order is important because information is lost when session IDs are the same in the two scans. When this occurs, by default the earlier session and vulnerability are overwritten with the later session and vulnerability. To prevent this when merging, you can choose another option for handling identical session IDs.

Note: Merging may work best with two scans that have few or no identical session IDs.

For all merge scan options, only sessions with an audit status of “Complete” in the source scan are merged. Session Exclusions (excluded from audit) are not merged. See "Audit settings: Attack Exclusions" on page 424 for more information.

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"
```

Note: 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.

Exit codes

The WI.exe application returns one of the exit codes described in the following table.

Code	Description
0	The command completed without errors.

Code	Description
-1 or -3	An error occurred.

Using WIScanStopper.exe

The WIScanStopper.exe application enables you to stop a scan that is currently running.

Note: This feature is not supported in OpenText DAST on Docker.

To stop a scan that is running, type the following on the command line:

```
WIScanStopper {scanid}
```

The WIScanStopper.exe application stops the scan with the specified scan ID (GUID). The application returns one of the exit codes described in the following table.

Code	Description
0	The scan successfully stopped.
1	The given argument is not a GUID. Try the command again with a valid scan ID (GUID).
2	The scan with the given GUID was not found to be running on the machine. Verify the scan ID (GUID) and try the command again.
3	A timeout occurred while waiting for the scan to stop. There is a 60 second timeout. When the stop command is sent, the process waits for the scan to stop. If 60 seconds elapses before the scan status changes, then the timeout occurs and the process returns this code.
4	Some other exception has occurred.

Tip: You can restart a scan that is stopped using the WI.exe application with the `-ir {scanid}` parameter. For more information, see "Options" on page 284.

Using MacroGenServer.exe

The MacroGenServer.exe application enables you to create a login macro from the command-line interface (CLI) by providing the start URL, username, and password. The following text provides sample syntax for using the application on the CLI:

```
macrogenserver.exe -u http://zero.webappsecurity.com/login.html -mu username -
mp password
```

Options

The available options are defined in the following table.

Parameter	Definition
`-u`	Specifies the start URL. This parameter is required.
`-mu`	Specifies the login form username. This parameter is required. **Important!** If the username contains special characters, you must wrap the string in double quotation marks. If the username contains the double quotation mark character, you must use the escape character to pass the quotation mark as part of the username. Refer to the documentation for the specific command-line interface you are using to determine the escape character.
`-mp`	Specifies the login form password. This parameter is required. **Important!** If the password contains special characters, you must wrap the string in double quotation marks. If the password contains the double quotation mark character, you must use the escape character to pass the quotation mark as part of the password. Refer to the documentation for the specific command-line interface you are using to determine the escape character.
`-m`	Specifies the file path where you want to save the login macro.
`-ps`	Identifies the IP address or host name of the proxy server. Examples: `macrogenserver.exe -u http://zero.webappsecurity.com -mu username -mp password -ps 127.0.0.1 -pp 8080` `macrogenserver.exe -u http://zero.webappsecurity.com -mu username -mp password -ps myproxyhostname -pp 8080`
`-pp`	Identifies the proxy server port.
`-at`	Specifies the network authentication type. Options are: • `Basic` • `Digest` • `Ntlm` • `ADFS_CBT`

Parameter	Definition
`-au`	Specifies the username for network authentication.
`-ap`	Specifies the password for network authentication.
`-h`	Displays the MacroGenServer application help.

Using the WISwag.exe tool

You can use the WISwag.exe tool in advanced situations for scanning a REST API, such as when you need to provide a configuration file that includes parameter values, overrides host information, or defines schemes (particularly for OData). The WISwag.exe tool is a command line tool that parses a REST API definition and converts it into a format that OpenText DAST understands.

Supported API definitions and protocols

The WISwag tool supports the following REST API definitions and protocols:

- Open API Specification versions 2.0 and 3.0 (formerly known as Swagger Specification). For more information, visit the Swagger website at http://swagger.io/.
- Open Data (OData) protocol (versions 2, 3, and 4). For more information, visit the OData website at http://www.odata.org/.

Tip: When using the WISwag tool with OData, if a POST fails to successfully create a request for an entity set, view the error in the HTTP details tab of the Web Macro Recorder to determine the requirements for the entity.

Locating the WISwag.exe tool

The WISwag.exe tool is included in the installation of OpenText DAST and is copied to the installation directory. By default, the installation directory is:

```
C:\Program Files\Fortify\Fortify WebInspect\
```

Process overview

The process for scanning a REST API is as follows.

Stage	Description
1.	Get the REST API definition from your development team.

Stage	Description
2.	Do one of the following: • If you do not have a settings file, use the WISwag.exe tool to convert the REST API definition into an OpenText DAST settings file. This option also generates a workflow macro and custom parameter rules, and embeds them in the settings file. See "Converting the API definition to a settings file" on page 308. • If you have a settings file, use the WISwag.exe tool to convert the REST API definition into an OpenText DAST workflow macro. See "Converting the API definition to a macro" on page 308.
3.	Use the webmacro or settings file to conduct a scan of your REST API.

WISwag.exe parameters

The WISwag.exe parameters are defined in the following table.

Parameter	Description
-a	Generates a JSON-formatted, human readable version of the API definition in the specified output file. The output file uses the `.json` extension. This parameter can be useful for debugging because the API definition is base64 encoded in the generated settings file. For more information, see "-s" on page 308. **Example:** `-a ./<api-def_filename>.json`
-ab	Passes the supplied authorization token as bearer-type authentication in the Authorization header. This parameter is applicable only if the API definition specifies “Authorization: Bearer” in the description. **Example:** `-ab QWxhZGRpbjpPcGVuU2VzYW1l`
-c	Generates custom parameter rules as a list of strings in the specified output file. The output file uses the `.txt` extension. The generated text file can be imported into the URL rewriting settings from the Advanced Settings in the Basic Scan Wizard. For more information, see "Scan settings: Custom Parameters" on page 382.

Parameter	Description
	Example output: `/odata-v4-test/Odata4Service.svc/Products({ID})` `/odata-v4-test/Odata4Service.svc/Categories({ID})`
-h	Generates HTTP requests for each audit session to be scanned in the specified output file. The output file uses the `.txt` extension. You can copy requests and paste them to the HTTP editor for debugging. **Example output:** `GET http://bhillwin7.spidynamics.com:8080/odata-v4-test/Odata4Service.svc/Products HTTP/1.1` `Accept: application/json;odata.metadata=full` `Host: bhillwin7.spidynamics.com:8080` `X-WISwag-ID: GET_/odata-v4-test/Odata4Service.svc/Products` `OData-Version: 4.0` `If-Match: *`
-i	Specifies the input file and location. The input file can be an API definition file or a configuration file. To override default settings and control which endpoints are processed, use a configuration file. For more information, see "Using a configuration file" on the next page. The location can be a URL or a local file. **Examples:** `-i http://mysite.com/api_def.json` `-i C:/myapi.json`
-it	Specifies the input type. Valid values are `odata` and `swagger`. **Examples:** `-it swagger` `-it odata`
-m	Generates an OpenText DAST macro in the specified output file. The output file uses the `.webmacro` extension. **Example:** `-m ./<macro_filename>.webmacro`

Parameter	Description
-ma	Injects the authorization header into the request for the API definition file. **Note:** This is useful if you use an authorization header in the configuration file and you need the same authorization header to be injected into the request for the API definition file.
-s	Generates an OpenText DAST settings file in the specified output file. The API definition along with any configuration overrides are added to the settings file. This is the recommended option when scanning a REST API. The output file uses the `.xml` extension. **Example:** `-s ./<settings_filename>.xml`

Converting the API definition to a macro

You can convert the API definition into an OpenText DAST workflow macro that you can then use to scan your REST API. To do this, enter the following command at the command line prompt:

```
WISwag.exe -it swagger –i http://<input_file_location> -m ./<macro_filename>.webmacro
```

Afterward, open the macro in the Web Macro Recorder tool and explore its contents.

Converting the API definition to a settings file

You can convert the API definition into an OpenText DAST settings file. The settings file is configured to run as Audit Only and contains a workflow macro and custom parameter rules derived from the REST API definition.

To do this, enter the following command at the command line prompt:

```
WISwag.exe -it swagger –i http://<input_file_location> -s ./<settings_filename>.xml
```

Open the scan settings in OpenText DAST and explore the contents. You should find that a workflow macro and custom parameter rules are already defined.

Using a configuration file

If you use a REST API definition file to create the workflow macro and settings file, then the macro and settings file will include only default values and settings. For more

advanced control over the HTTP requests generated by the WISwag tool, you can pass a configuration file to the WISwag tool instead of a REST API definition. This advanced configuration is useful in cases where control over specific operations or parameters is required. For example, you might need to exclude certain operations, such as logout or delete operations, from an OpenText DAST scan. You can accomplish this by listing the operation IDs in the '`excludeOperations`' property. Operation IDs are defined in the REST API definition. Sometimes an allow-list approach is easier when only a few operations need to be tested. In this case, use the '`includeOperations`' list.

For more information, see "Understanding the API scan configuration file" on page 157.

Configuration file format

The configuration file has the following format:

```
{
 apiDefinition : 'http://mysite.com/api_def.json', /* can also be a local
file (ex. C:/myapi.json) */
 host : 'localhost:8080', /* replace the host in every generated request */
 schemes : ['https', 'http'], /* generate output for both of these schemes
*/
 preferredContentType : 'application/json', /* if given a choice, prefer
json */
 excludeOperations : [ 'logoutUser', 'deleteUser' ], /* generate no output
for these operations */
 parameterRules :
 [
  {
   name : 'userId',
   value : 42,
   location : 'path',
   type : 'number',
   includeOperations : ['createNewUser', 'getUser'] /* only apply this rule
to these operations */
  },
  {
   name : 'file',
   value : 'my file payload',
   filename : 'myfile.txt',
   location : 'body',
   type : 'file'
  },
  {
   name : 'Authorization',
```

```
    value : 'Basic QWxhZGRpbjpPcGVuU2VzYW1l',
    location : 'header',
    inject : true /* add this header to every generated request */
   }
  ]
}
```

Regular expressions

Special metacharacters and sequences are used in writing patterns for regular expressions. The following table describes some of these characters and includes short examples showing how the characters are used. Another recommended resource is the Regular Expression Library at http://regexlib.com/Default.aspx.

Tip: To verify the syntax of regular expressions you create, use the Regular Expression Editor (if it is installed on your system).

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

Character	Description
\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 line feed character.
\r	Matches a carriage return character.
\s	Matches any white space including space, tab, form-feed, and so on. Equivalent to [\f\n\r\t\v]
\S	Matches any nonwhite space character. Equivalent to [^ \f\n\r\t\v]
\w	Matches any word character including underscore. Equivalent to [A-Za-z0-9_].
\W	Matches any nonword character. Equivalent to [^A-Za-z0-9_].

OpenText DAST engineers have also created and implemented extensions to the normal regular expression syntax. For more information, see "Regex extensions" below.

Regex extensions

OpenText engineers have developed and implemented extensions to the normal regular expression (regex) syntax. When building a regular expression, you can use the tags and operators described below.

Regular expression tags

- [STATUSCODE]
- [BODY]
- [ALL]
- [URI]
- [HEADERS]

- [COOKIES]
- [STATUSLINE]
- [STATUSDESCRIPTION]
- [SETCOOKIES]
- [METHOD]
- [REQUESTLINE]
- [VERSION]
- [POSTDATA]
- [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: You must include a space (ASCII 32) before and after an "open" or "close" parenthesis; otherwise, the parenthesis will be erroneously considered as part of the regular expression.

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

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

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

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

See also

"Regular expressions" on page 310

OpenText DAST REST API

This topic provides information about the OpenText DAST REST API.

What is the OpenText DAST REST API?

The OpenText DAST REST API provides a RESTful interface between your systems and OpenText DAST for remotely controlling the proxy and scanner. It runs as a lightweight Windows service (named WebInspect API) that is installed automatically when you install OpenText DAST. You configure, start, and stop the service using the Fortify Monitor tool. You can use the OpenText DAST REST API to add security audit capabilities to your existing automation scripts.

The OpenText DAST REST API is fully described and documented using the industry-standard Swagger RESTful API Documentation Specification version 2.0 (now known as OpenAPI Specification). The Swagger documentation provides detailed schema, parameter information, and sample code to simplify consumption of the REST API. It also provides functionality for testing the endpoints before using them in production.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Configuring the OpenText DAST REST API

Before you can use the OpenText DAST REST API, you must configure it.

1. From the Windows Start menu, click **All Programs > OpenText > OpenText DAST Monitor**.
   The OpenText DAST Monitor icon appears in the system tray.
2. Right-click the **OpenText DAST Monitor** icon, and select **Configure DAST API**.
   The Configure DAST API dialog box appears.
3. Configure the API Server settings as described in the following table.

Setting	Value
Host	Both OpenText DAST and the OpenText DAST REST API must reside on the same machine. The default setting, +, is a wild card that tells the OpenText DAST REST API to intercept all request on the port identified in the Port field. If you have another service running on the same port and want to define a specific hostname just for the API service, this value can be changed.
Port	Use the provided value or change it using the up/down arrows to an available port number.
Authentication	Choose **None**, **Windows**, **Basic**, or **Client Certificate** from the Authentication drop-down list. If you choose **Basic** for authentication, you must provide user name(s) and password(s). To do this: a. Click the **Edit passwords** button and select a text editor. The `wircserver.keys` file opens in the text editor. The file includes sample user name and password entries: `username1:password1` `username2:password2` b. Replace the samples with user credentials for access to your server. If additional credentials are needed, add a user name and password, separated by a colon, for each user to be authenticated. There should be only one user name and password per line. c. Save the file. If you choose **Client Certificate** for authentication, you must

Setting	Value
	first generate a client certificate based on your root SSL certificate issued by a trusted certificate authority (CA), and then install it on the client machine. **Tip:** You can use a tool, such as the MakeCert utility in the Windows Software Development Kit (SDK), to create your client certificate.
Use HTTPS	Select this check box to access the server over an HTTPS connection. To run the server over HTTPS, you must create a server certificate and bind it to the API service. To quickly create a self-signed certificate to test the API over HTTPS, run the following script in an Administrator PowerShell console: `$rootcertID = (New-SelfSignedCertificate -DnsName "DO NOT TRUST - WIRC Test Root CA","localhost", "$($env:computername)" -CertStoreLocation "cert:\LocalMachine\My").Thumbprint` `$rootcert = (Get-Item -Path "cert:\LocalMachine\My\$($rootcertID)")` `$trustedRootStore = (Get-Item -Path "cert:\LocalMachine\Root")` `$trustedRootStore.open("ReadWrite")` `$trustedRootStore.add($rootcert)` `$trustedRootStore.close()` `netsh http add sslcert ipport=0.0.0.0:8443 certhash=$($rootcertID) appid="{160e1003-0b46-47c2-a2bc-01ea1e49b9dc}"` The preceding script creates a certificate for the local host and the computer name, puts the certificate in the Personal Store and Trusted Root, and binds the certificate to port 8443. If you use a different port, specify the port you use in the script. **Important!** Use the self-signed certificate created by the preceding script for testing only. The certificate works only on your local machine and does not provide the security of a certificate from a certificate authority. For production, use a certificate that is generated by a certificate authority.

Setting	Value
Log Level	Choose the level of log information you want to collect. **Tip:** You can view the API log files using the Windows Event Viewer. The log files are located under **Applications and Services Logs > WebInspect API**.

4. Do one of the following:
   - To start the OpenText DAST REST API service and test the API configuration, click **Test API**.

     The service starts, and a browser opens and navigates to the OpenText DAST REST API Swagger UI page. For more information about this page, see "Accessing the OpenText DAST API Swagger UI" below.
   - To start the OpenText DAST REST API service without testing the API configuration, click **Start**.

     Important! If you change the default settings and close the Configure DAST API dialog box without starting the service, a prompt advising of unsaved changes appears. To save the changes to the settings configuration file, click **Yes**. Otherwise, the changes will be discarded.

Accessing the OpenText DAST API Swagger UI

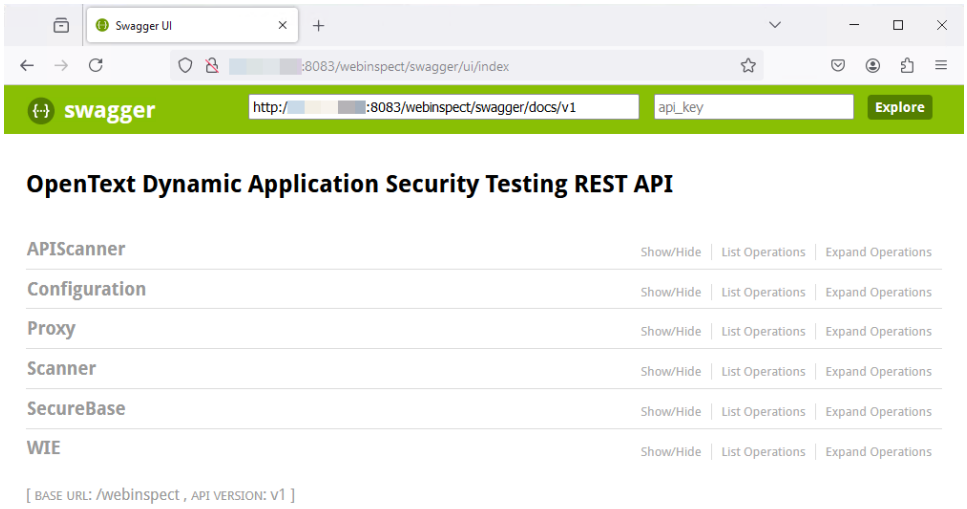
Complete documentation—including detailed schema, parameter information, sample code, and functionality for testing endpoints—is included in the OpenText DAST API Swagger UI.

To access this information:

1. After configuring and starting the OpenText DAST API service, open a browser.
2. Type `http://<hostname>:<port>/webinspect/api` in the address field and press **Enter**.

   Example: If you used the default settings when configuring the OpenText DAST REST API, you would type `http://localhost:8083/webinspect/api`.

The OpenText DAST REST API Swagger UI page appears.



Switching between API versions

By default, the Swagger UI opens to version 1 (v1) of the OpenText DAST REST API. Version 2 (v2) of the API includes asynchronous versions of endpoints that take a long time to complete. Using these endpoints generates a job token that you can use with the v2 `Job` endpoints to get the status and results from the job.

To switch to OpenText DAST REST API v2:

- In the Swagger UI, change **v1** at the end of the URL to **v2**.



To switch back to OpenText DAST REST API v1:

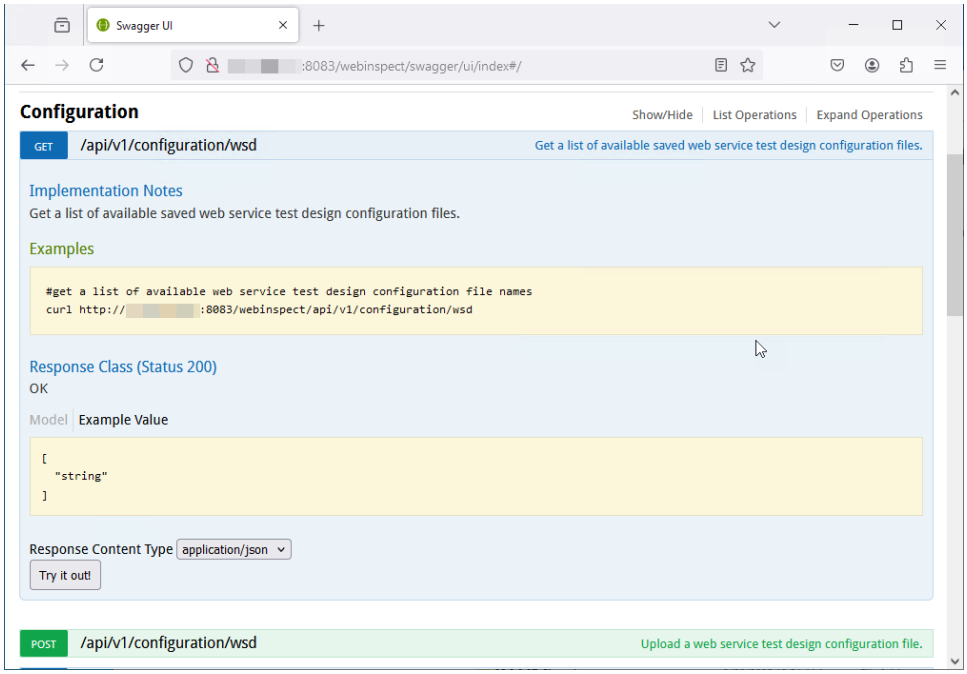
- In the Swagger UI, change **v2** at the end of the URL to **v1**.

Using the Swagger UI

To use the Swagger UI:

1. In the Swagger UI, click an endpoint category.
2. Click the endpoint method to use.

Detailed schema, parameter information, sample code, and functionality for testing the endpoint appear.

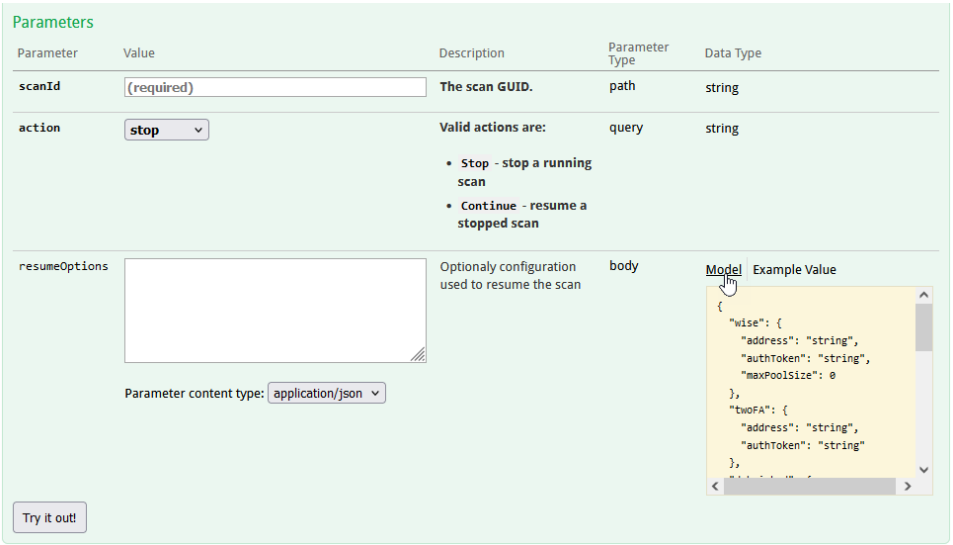


Getting field-level details

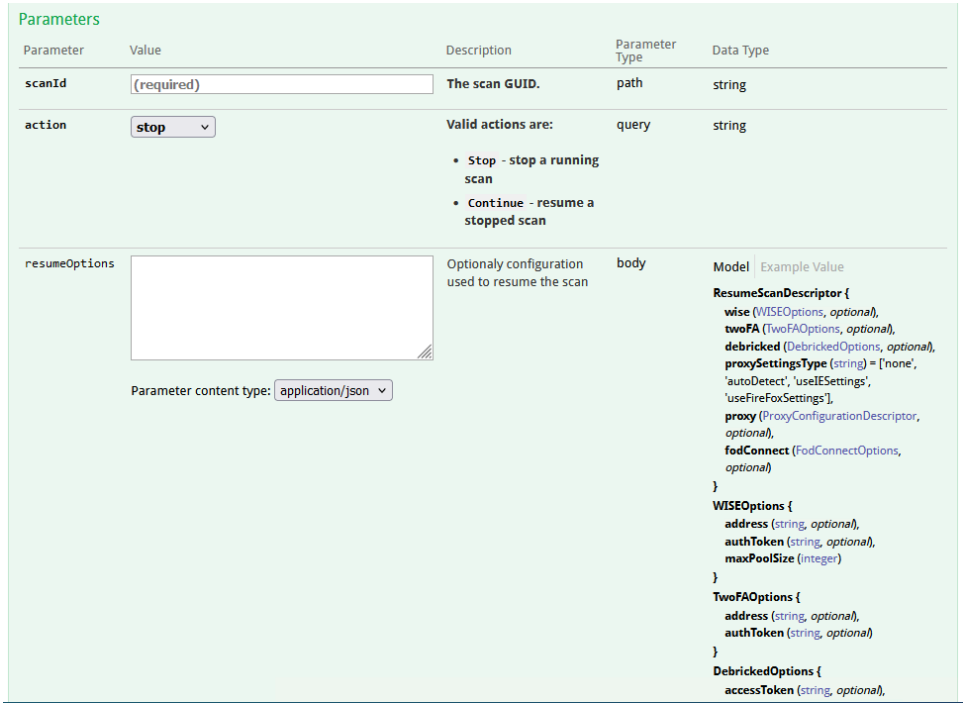
Some API endpoints have numerous fields that you can configure. These fields are documented in detail in the Swagger UI.

To view the field-level details:

- In the **Parameters** section of the endpoint, click **Model** under the **Data Type** heading.



Additional details for all the endpoint fields appear.



Automating OpenText DAST

You can use the OpenText DAST API to add OpenText DAST to your existing automation scripts. As long as the user agent can access the Service Router, the scripts can exist in an entirely different environment from OpenText DAST.

OpenText DAST updates and the API

After updating OpenText DAST, you must open the OpenText DAST user interface and then open a scan so that any database schema changes can be applied to the scan database. Otherwise, you may not be able to run certain API commands without receiving an error.

Scanning with a Postman collection

You can use your existing Postman automation test scripts, also known as collections, to conduct scans of REST API applications. This topic provides general information about Postman and the additional third-party software that is required.

What is Postman?

Postman is an API development environment that enables you to design, collaborate on, and test APIs. Postman lets you create collections for your API calls, where each

collection can be organized into subfolders and multiple requests. You can import and export collections, making it easy to share files across your development and testing environment. Through the use of a Collection Runner such as Newman, tests can be run in multiple iterations, saving time on repetitive tests.

Benefits of a Postman collection

A REST API application does not expose all the endpoints in a format that a human with a browser or an automated tool can consume. It is often simply a collection of endpoints that accepts various posts, puts, and gets with a specific set of request data. To successfully audit these endpoints, OpenText DAST needs to understand key details about the API. A well-defined Postman collection can expose these endpoints so that OpenText DAST can audit the API application.

Known limitations with Postman variables

OpenText DAST does not support Data variables in Postman. However, it does support Collection, Global, and Environment variables, as well as Local variables in a collection.

As a workaround, you can specify Data variables in an Environment, which is a set of variables that you can use in your Postman requests.

Options for Postman scans

You can conduct a Postman scan using one of following options:

- API Scan Wizard (See "Using the API Scan Wizard" on page 137)
- WI.exe or the OpenText DAST REST API (See "Postman API scan using WI.exe or OpenText DAST REST API" on page 326)

Postman prerequisites

A Postman collection version 2.0 or 2.1 is required for conducting scans in OpenText DAST. Additionally, you must install Newman command-line collection runner, Node.js, and Node Package Manager (NPM). For specific version information and additional instructions, see the *OpenText™ Application Security Software System Requirements*.

Using client certificates with Postman

To use a client certificate as authentication for a Postman scan, the certificate file format must be supported by Windows. If the client certificate is not Windows-compatible, you can convert the certificate to a Windows-compatible format and then use the converted file for your Postman scan.

The following table describes the process for converting and using a client certificate with Postman.

Stage	Description
1.	Use a tool such as OpenSSL to convert the certificate to a Windows format.
2.	Install the converted certificate in the Windows certificate store on the machine where OpenText DAST is installed.
3.	Add the certificate to the Scan Settings: Authentication. For more information, see "Scan settings: Authentication" on page 393.

Tips for preparing a Postman collection

This topic provides tips for creating a good Postman collection.

Ensure valid responses

In order to get valid responses, the collection must be complete and executable. Requests must include:

- A valid request URL
- The correct HTTP method (POST, GET, PUT, PATCH, or DELETE)
- Valid parameter data that allows proper exercising of the API

  For example, if you have a “name” parameter, then you must provide actual sample data such as "King Lear" or "Hamlet," rather than the default data type “string.”

Order of requests

Remember that the order of operations or requests is important. For example, you must create (or POST) sample data to a parameter before you can do a GET or a DELETE operation on the data.

Tip: To avoid URL errors while running the collection in OpenText DAST, after bundling the API requests in the correct order in your collection, save each request individually by clicking the request and then clicking **Save**.

Handling authentication

If your API requires authentication, you must configure it in the Postman collection. Follow these guidelines when configuring authentication:

- The user credentials must be current and not expired.
- If you use an environment to specify authentication information, select the type of authentication environment in the Postman collection.
- It is possible that not all requests in the collection require authentication or not all requests require the same type of authentication. If this is the case in your collection, be sure to specify the appropriate authentication type for each request in the

collection.

Important! If session state is lost while using various authentication types in a scan, it will not be restored correctly. For proper restoration of session state, use a login macro or Postman login collection with a single type of authentication.

Using static authentication

When using static authentication, you must hard-code user credentials as a name/value pair in the Postman collection. When OpenText DAST parses the collection file, it determines the type of authentication being used and retrieves the key name and value from the collection. These values are then added to the scan settings.

OpenText DAST supports the following types of static authentication:

- API Key
- Basic
- Bearer Token
- Digest
- NTLM
- Oauth 1.0
- Oauth 2.0

Using dynamic authentication

When using dynamic authentication, you must store the Bearer token or API key authentication variables in either a Postman environment file or a collection file. For example, a Bearer Token may use a variable such as `{{bearerToken}}`.

You must use regular expressions in a response state rule to dynamically supply the Bearer token or API key during the scan. The response state rule provides search and replace options that enable the token or key to be retrieved from a response and then used in future sessions. For more information, see "Scan settings: HTTP Parsing" on page 376.

Using a Postman login macro

You can provide a login macro and a workflow macro in the form of Postman collection files in the OpenText DAST REST API or Wi.exe. For example, you can specify a login macro file such as `LoginBearer.json`. When using a login macro, however, you must also specify a logout condition, such as the regular expression `The\stoken\sis\snot\svalid`.

Postman auto-configuration

Auto-configuration for static authentication is supported when the authentication values are known, such as when the username and password are hard-coded in the authentication section of the collection. If auto-configuration is not disabled, OpenText

DAST checks the authentication portion of the collection file for valid values that are then applied to the scan settings.

Auto-configuration for dynamic authentication attempts to automatically provide a login macro and response state rule. It is useful when the Bearer token or API key is stored in a variable. If successful, a message indicates that authentication for Postman collection was detected. If a Bearer token was detected but a stable configuration was not created, a message indicates that auto-configuration failed and provides the reason.

Important! Auto-configuration for dynamic authentication works only for simple cases using Bearer token authentication.

If auto-configuration fails, you must manually configure authentication. For more information, see "Manually configuring Postman login for dynamic tokens" below.

Sample Postman scripts

Sample code for leveraging the Postman API can be found at https://github.com/fortify/WebInspectAutomation.

A sample Postman collection is available for download on the Fortify repository on GitHub at https://github.com/fortify/WebInspectAutomation/tree/master/PostmanSamples.

Manually configuring Postman login for dynamic tokens

This topic describes how to configure dynamic authentication manually if auto-configuration fails for a Postman scan. Dynamic authentication uses dynamic tokens.

What are dynamic tokens?

Dynamic tokens are authentication tokens that are generated by software and are unique for each instance of authentication. Tokens can be created for a short period of time, and each instance is renewed individually.

Before you begin

You must know the following to configure manual login:

- The type of authentication used in your application (such as Bearer, API key, OAuth1.0, OAuth 2.0, Cookie)
- How to create regular expression search arguments

Process overview

The process to manually configure login is described in the following table.

Stage	Description
1.	Identify and isolate the login request or requests in a separate Postman collection. For more information, see "Identifying and isolating the login request" below.
2.	Create a logout condition regular expression. For more information, see "Creating a logout condition with regular expressions" below.
3.	Create a response state rule. For more information, see: • "Creating a response state rule for a bearer token" on the next page • "Creating a response state rule for an API key" on the next page **Note:** A response state rule is not needed for cookie session management.

Identifying and isolating the login request

To identify and isolate the login request:

1. Examine the Postman collection contents to identify the login request.

   Tip: Typically, the login request is the first request in the Postman collection that obtains an authentication token. However, authentication could involve several requests.

2. Copy this request or multiple requests.
3. Paste the request(s) in a separate file.
4. Save the file as a Postman collection.

Creating a logout condition with regular expressions

To create a logout condition:

1. Find several requests that require authentication.
2. Do one of the following:
   - For a bearer token, replace the auth token with an incorrect value and send it to the application.
   - For an API key, send an incorrect APIKey value to the application.
3. Use the reply from these requests to create a regular expression that matches these responses and does not match a valid session.

   For example, if you see the word “unauthorized” in most cases, then it is the best word to use in the regular expression, such as:

```
[STATUSCODE]200 AND [BODY]unauthorized
```

If an incorrect APIKey value gets a reply of “{"status": "Access Deny"}”, then the best regular expression would be:

```
[BODY]Access\sDeny
```

Creating a response state rule for a bearer token

To create a response state rule for a bearer token, you must create two regular expressions.

The first regular expression searches all responses for an authentication token update. Typically, this token will be in response to the login request that was identified in Stage 1 of the process.

For example, in the following response, we see a reference to "token."

```
"{"success":true,"message":"Authentication
      successful!","token":"eyJhbGciOiJIUzI1NiIs
      InR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluI
      iwiaWF0IjoxNTg1NzQzNzkzLCJleHAiOjE1ODU3NDc
      zOTN9.i8uXa20JQt00tlOjd1twRD76jTnsG-0xiU97
      QWy6jkg"}"
```

For this response, we can create the following regular expression:

```
"token":"(?<Token>[-a-zA-Z0-9._~+/]+?=*)"}$
```

In this regular expression, the `(?<Token>[-a-zA-Z0-9._~+/]+?=*)` identifies the value of the token.

Note: XML uses character escaping. When you use regular expressions that include < and > symbols in XML format, the `<` symbol escapes with `<` and the `>` symbol escapes with `>`.

The second regular expression indicates where to store this token. For a bearer token, it will be in the “Authorization: Bearer ….” header.

The following is an example for a bearer token:

```
Authorization:\sBearer\s(?<Token>[^\r\n]*)\r\n
```

In this second regular expression, the `(?<Token>[^\r\n]*)` identifies the value that should be replaced with the value from the first regular expression.

Creating a response state rule for an API key

To create a response state rule for an API key, you must create two regular expressions.

The first regular expression searches all responses for an authentication token update. Typically, this token will be in response to the login request that was identified in Stage 1 of the process.

For example, assume that you have a header API key type of auth. A request sends the username and password to the path “/Login” and returns a response similar to the following:

```
"{"success":true,"APIToken":
      "tp8989ieupgrjynsfbnfgh9ysdopfghsprohjo"}"
```

All protected requests send an “APIKey: ….” header to authorize access.

For this response, we can create the following regular expression:

```
"APIToken":"(?<APIToken>[a-zA-Z0-9]+?)"}$
```

Note: XML uses character escaping. When you use regular expressions that include < and > symbols in XML format, the < symbol escapes with `<` and the > symbol escapes with `>`.

The second regular expression indicates where to store this token. For an APIKey, it could be a custom header name and value or a custom query parameter name and value.

```
APIKey:\s(?<APIToken>[^\r\n]*)\r\n
```

Postman API scan using WI.exe or OpenText DAST REST API

This topic describes the process for conducting a scan using a Postman collection in the OpenText DAST REST API or Wi.exe. To conduct a scan using the API Scan Wizard, see "Using the API Scan Wizard" on page 137.

Recommendation

OpenText recommends that you run only one scan at a time. When using SQL Express, in particular, depending on the size of the site, conducting concurrent (or parallel) scans might result in high usage of RAM, CPU, and disk resources on the OpenText DAST host.

Process

The following table describes the process for conducting a scan using a Postman collection.

Stage	Description
1.	Do the following in Postman: 1. Create a Postman collection file, following the guidelines mentioned previously in this topic. 2. Save each API call in Postman individually. 3. Click **Runner** to open the Newman command-line Collection Runner.
2.	Do the following in Newman command-line Collection Runner: 1. With the collection open in the Collection Runner, ensure that the API calls are in the correct order for execution. 2. Click **Run *<Collection Name>***. 3. Inspect the responses from each call to ensure the requests were successful.
3.	Do one of the following in OpenText DAST: • To use the OpenText DAST REST API: a. Configure and start the OpenText DAST API. See "Configuring the OpenText DAST REST API" on page 314. b. Access the Postman API endpoint in the Swagger UI. See "Accessing the OpenText DAST API Swagger UI" on page 316. c. Configure the endpoint according to the instructions in the Swagger UI. d. Execute the endpoint sample scripts from the Swagger UI or your API tool of choice. **Important!** Include a scan settings file with the appropriate settings that provide access to the site in your Postman collection. For example, include the correct allowed hosts, proxy settings, and so on. If you do not specify a settings file, then the default scan settings from OpenText DAST are applied to the scan. • To use Wi.exe: a. Launch the CLI as described in "Command-line execution" on

Stage	Description
	page 283. b. Construct your command using the Postman Scans options described in "Using wi.exe" on page 284.
4.	The endpoint or CLI command returns the scan ID (GUID) and the results of the Postman collection.

Troubleshooting a Postman scan

Use the following troubleshooting tips if you encounter issues while running a scan with a Postman collection:

1. Check the proxy configuration in the scan settings to ensure that Postman can run through the proxy and access the site for testing. One option is to try running Newman manually with the proxy configuration.
2. Check the results of the request:
   a. View the total requests sent to ensure that they match the requests in the Postman file.
   b. Ensure that there are no failed requests.

Integrating with Selenium WebDriver

You can integrate OpenText DAST with Selenium WebDriver, also known as Selenium 2.0, to do the following:

- Conduct a scan using the `WI.exe` command-line tool
- Create a workflow macro using the OpenText DAST REST API

Known limitations

The following are known limitations for integrating OpenText DAST with Selenium Webdriver:

- OpenText DAST supports Selenium WebDriver only.
- OpenText DAST does not support Selenium WebDriver with remote server configuration, such as the RemoteWebDriver class.
- A Selenium WebDriver macro can be used as a workflow macro only. It cannot be a login or startup macro.
- You can initiate a scan using a Selenium WebDriver macro from the command line interface (CLI) or the API only. While you cannot initiate a scan from the user interface, you can rescan and import/export a Selenium WebDriver macro.

Process overview

The process for integrating OpenText DAST with Selenium WebDriver is described in the following table.

Stage	Description
1.	OpenText DAST must be able to capture traffic from a web browser using the OpenText DAST proxy. Do one of the following to enable proxy capture: • Add the proxy to your Selenium scripts directly in the code or using a placeholder in the command line interface as described in "Adding the proxy to Selenium scripts" on the next page. • Use the OpenText DAST geckodriver.exe for capturing traffic when using Firefox as described in "Using the OpenText DAST geckodriver.exe" on page 335.
2.	Install the Selenium WebDriver environment on the machine running OpenText DAST as described in "Installing the Selenium WebDriver environment" on page 335.
3.	Ensure that you can start up the Selenium WebDriver scripts from the command line and define your Allowed Hosts as described in "Testing from the command line" on page 336.
4.	Optionally, upload all scripts and their dependencies to the Selenium API or manually copy them to the machine running OpenText DAST as described in "Uploading files to OpenText DAST" on page 339.
5.	Use the command from Stage 3 to run a scan using WI.exe or create a macro using the OpenText DAST REST API as described in "Using the Selenium command" on page 340.
6.	Fix any errors that occur. When conducting a scan with WI.exe or creating a macro in the API, the macro is validated. Errors and warnings are returned for each Selenium command. This feature is enabled by default. To disable it: • In WI.exe, use the argument `-selenium_no_validation` parameter. For more information, see "Using wi.exe" on page 284. • In the API, set the `VerifyMacro` parameter to `false`. For more information, see the OpenText DAST REST API Swagger UI.

Stage	Description
	To troubleshoot issues, view the Scan logs for errors and the StateRequestor logs for warnings. **Tip:** Generally, logs are written to the following directory paths: • If an API scan runs as the SYSTEM USER, which is the default user, then logs are written to: `C:\ProgramData\HP\HP WebInspect\Schedule\logs\<scan_guid>\ScanLog` `C:\ProgramData\HP\HP WebInspect\Schedule\logs\<scan_guid>\StateRequestor` • All CLI and UI scans, and if an API scan runs as the current user, then logs are written to: `C:\Users\<user.name>\AppData\Local\HP\HP WebInspect\Logs\<scan_guid>\ScanLog` `C:\Users\<user.name>\AppData\Local\HP\HP WebInspect\Logs\<scan_guid>\StateRequestor`

Adding the proxy to Selenium scripts

To use this method of capturing traffic from the web browser, you must add OpenText code that applies the proxy to your Selenium initialization directly in your code or, if applicable, passed as an argument in the command line interface (CLI).

Advantages

This approach provides flexibility, as it can run from any browser that Selenium supports. Additionally, this approach should provide some upgrade protection. The OpenText code resides in your scripts, so you should be able to continue using it in future versions of Selenium with only minor code changes.

Disadvantages

This approach involves a one-time manual task of adding OpenText code to your scripts for initializing the browser correctly.

Sample code

You must get the value from the environmental variable named `Fortify_WI_Proxy`, and then store it as an HTTP and HTTPS proxy for the web browser and trust certificate. How you do this depends on your programming language. The following sections provide sample code for several languages.

Note: These code samples are based on Selenium WebDriver version 3.14. Code for your specific version might be different.

C#

In your C# code, you must find where the browser driver is initialized and add browser options to it. The following is an example for the Chrome browser.

```
     ChromeOptions chromeOptions = new ChromeOptions();

     string proxy = Environment.GetEnvironmentVariable("Fortify_WI_Proxy");
     if (!String.IsNullOrEmpty(proxy))
     {
          chromeOptions.AcceptInsecureCertificates = true;
          chromeOptions.Proxy = new Proxy();
          chromeOptions.Proxy.HttpProxy = proxy;
          chromeOptions.Proxy.SslProxy = proxy;
     }
     …. new ChromeDriver(chromeOptions) // options should go into this
class
```

The following is an example for the Firefox browser.

```
     FirefoxOptions config = new FirefoxOptions();

     string proxy = Environment.GetEnvironmentVariable("Fortify_WI_Proxy");
     if (!String.IsNullOrEmpty(proxy))
     {
          config.AcceptInsecureCertificates = true;
          config.Proxy = new Proxy();
          config.Proxy.HttpProxy = proxy;
          config.Proxy.SslProxy = proxy;
     }
     … new FirefoxDriver(config))
```

Java

In your Java code, you must find where the browser driver is initialized and add browser options to it. The following is an example for the Chrome browser.

```
ChromeOptions options = new ChromeOptions();
String wi_proxy = System.getenv("Fortify_WI_Proxy");
if (wi_proxy != null) {
    Proxy proxy = new Proxy();
```

```
    proxy.setHttpProxy(wi_proxy);
    proxy.setSslProxy(wi_proxy);
    options.setProxy(proxy);
    options.setAcceptInsecureCerts(true);
}

ChromeDriver driver=new ChromeDriver(options);
```

The following is an example for the Firefox browser.

```
FirefoxOptions options = new FirefoxOptions();
String wi_proxy = System.getenv("Fortify_WI_Proxy");
if (wi_proxy != null) {
    Proxy proxy = new Proxy();
    proxy.setHttpProxy(wi_proxy);
    proxy.setSslProxy(wi_proxy);
    options.setProxy(proxy);
    options.setAcceptInsecureCerts(true);
}

FirefoxDriver driver=new FirefoxDriver(options);
```

JavaScript

In your JavaScript code, you must find where the browser driver is initialized and add browser options to it. The following is an example for the Chrome browser.

```
const selProxy = require('selenium-webdriver/proxy');
……
(async function example() {
  let env =    process.env.Fortify_WI_Proxy;
  if (env) {
         let caps = { acceptInsecureCerts: true }; //allow to accept all
certificates
         let proxy = { http: env, https: env };  // apply env variable as
proxy
         driver = await new Builder().withCapabilities(caps).setProxy
(selProxy.manual(proxy)).forBrowser('chrome').build(); // set proxy and
acceptInsecureCerts
  }else
         driver = await new Builder().forBrowser('chrome').build();
```

The following is an example for the Firefox browser.

```
const selProxy = require('selenium-webdriver/proxy');
……
  let env =    process.env.Fortify_WI_Proxy;
  if (env) {
         let caps = { acceptInsecureCerts: true };  //allow to accept all
certificates
         let proxy = { http: env, https: env };  // apply env variable as
proxy
         driver = await new Builder().withCapabilities(caps).setProxy
(selProxy.manual(proxy)).forBrowser('firefox').build(); // set proxy and
acceptInsecureCerts
  }else
         driver = await new Builder().forBrowser('firefox').build();
```

Python

In your Python code, you must find where the browser driver is initialized and add browser options to it. The following is an example for the Chrome browser.

```
capabilities1 = DesiredCapabilities.CHROME.copy()
Fortify = os.environ.get('Fortify_WI_Proxy')
if Fortify is not None:
    prox = Proxy()
    prox.proxy_type = ProxyType.MANUAL
    prox.http_proxy = Fortify
    prox.ssl_proxy = Fortify
    prox.add_to_capabilities(capabilities1)
cls.driver = webdriver.Chrome(executable_path='C:/chromedriver.exe',
desired_capabilities=capabilities1)
```

The following is an example for the Firefox browser.

```
import os
from selenium.webdriver import DesiredCapabilities
from selenium.webdriver.common.proxy import Proxy, ProxyType
……
capabilities1 = DesiredCapabilities.FIREFOX.copy()
Fortify = os.environ.get('Fortify_WI_Proxy')
if Fortify is not None:
    capabilities1['acceptInsecureCerts'] = True
    prox = Proxy()
    prox.proxy_type = ProxyType.MANUAL
```

```
    prox.http_proxy = Fortify
    prox.ssl_proxy = Fortify
    prox.add_to_capabilities(capabilities1)
cls.driver = webdriver.Firefox(executable_path='C:/geckodriver.exe',
capabilities=capabilities1)
```

Ruby

In your Ruby code, you must find where the browser driver is initialized and add browser options to it. The following is an example for the Chrome browser.

```
http_proxy = ENV['Fortify_WI_Proxy']

if http_proxy
       proxy = Selenium::WebDriver::Proxy.new(http: http_proxy,
ssl: http_proxy)
       capabilities = Selenium::WebDriver::Remote::Capabilities.chrome
(accept_insecure_certs: true)
       capabilities.proxy = proxy;
else
    capabilities = Selenium::WebDriver::Remote::Capabilities.chrome()
end

driver = Selenium::WebDriver.for :chrome, desired_capabilities:
capabilities
```

The following is an example for the Firefox browser.

```
http_proxy = ENV['Fortify_WI_Proxy']

if http_proxy
       proxy = Selenium::WebDriver::Proxy.new(http: http_proxy,
ssl: http_proxy)
       capabilities = Selenium::WebDriver::Remote::Capabilities.firefox
(accept_insecure_certs: true)
       capabilities.proxy = proxy;
else
    capabilities = Selenium::WebDriver::Remote::Capabilities.firefox()
end

driver = Selenium::WebDriver.for :firefox, desired_capabilities:
capabilities
```

Using the CLI

If your scripts accept an argument that configures the proxy, then you can use this method to add the OpenText DAST proxy to your scripts. For example, if you have an argument named `-proxy "<host:port>"`, then you can use the placeholder `{Fortify_WI_Proxy}` in the command at run time as shown here:

```
-proxy "{Fortify_WI_Proxy}"
```

If you must specify the host and port separately, then you can use a placeholder for each as shown here:

```
-proxy "{Fortify_WI_Proxy_Host}:{Fortify_WI_Proxy_Port}"
```

These arguments will replace the placeholder in your scripts with the OpenText DAST proxy at run time.

Using the OpenText DAST geckodriver.exe

GeckoDriver is a proxy that helps W3C WebDriver-compatible clients communicate with Gecko-based browsers. The `geckodriver.exe` application provides this proxy for Firefox browsers. To use this method of capturing traffic from the web browser, you must replace your existing `geckodriver.exe` with the OpenText DAST `geckodriver.exe`, which you can find in the `<InstallationDirectory>\Extensions` folder.

Note: The default installation directory is `C:\Program Files\Fortify\Fortify WebInspect\Extensions`.

Advantages

This approach requires less work for you.

Disadvantages

You will not be able to use the latest version of geckodriver.exe, and you must use only Firefox scripts.

Installing the Selenium WebDriver environment

On the machine where OpenText DAST is installed, you must install all the software and tools that you need to run Selenium scripts. This includes, but is not limited to, such items as:

- A browser
- A test runner

- All prerequisite software to support running Selenium scripts

  For example, for .NET NUnit framework, you must install `.NET` and `nunit3-console.exe` as the executable that runs the Selenium scripts.

Important! The list of required software and tools varies depending on your programming language.

Testing from the command line

To ensure that you are able to start up and run Selenium WebDriver scripts from the command line, you must create and use a command that will execute your Selenium script. The command that you use varies depending on the programming language and testing framework that you are using to conduct Selenium tests.

For example, to run NUnit in .NET, you can run a command similar to the following:

```
D:\tmp\selenium_wd\bin\net35\nunit3-console.exe "D:\tmp\selenium_
wd\selenium_c_sharp-master\Selenium\bin\Debug\Selenium.dll"
```

In this example, the `nunit3-console.exe` is the unit test runner, and `Selenium.dll` is the DLL that contains the unit tests. For more examples, see "Creating a Selenium command" below.

Tip: You can use the `POST /configuration/selenium/folder` and `GET /configuration/selenium/file/{foldername}` API endpoints to show the full path to the files you deployed. You can use this information to update the command in the CLI. For more information, see "Uploading files to OpenText DAST" on page 339.

Creating a Selenium command

The Selenium command is used on the command line to execute unit tests. In most cases, the command can be found during a run of unit tests on the build server or while debugging. This command varies based on the unit test framework that you are using. Each framework has its own runner and command-line arguments. The following sections provide tips and sample commands for several frameworks in various languages.

.NET MSTest

The MSTest framework uses a tool called `Vstest.console.exe` with the following syntax:

<Path_to_Vstest_Executable>\Vstest.console.exe *<Path_to_Unit_Test_dlls>**<TestFileNames>* *<Options>*

In most cases, you must call this executable with a list of DLLs, which are the test file names that you want to run. The following sample code runs two test files:

```
"C:\Program Files\Microsoft Visual Studio 14.0\Common7\IDE\
   CommonExtensions\Microsoft\TestWindow\vstest.console.exe"
   "C:\Projects\Tests\bin\TestHomepage_unittest.dll"
   "C:\Projects\Tests\bin\AddCart_unittest.dll"
```

.NET NUnit

The NUnit framework uses a tool called `nunit3-console.exe` (version 3.*x*) with the following syntax:

`NUNIT3-CONSOLE <InputFiles> <Options>`

You must call this executable with a list of DLLs, which are the test file names that you want to run. The following sample code runs two test files:

```
C:\nunit\net35\nunit3-console.exe
   "C:\Projects\Tests\bin\TestHomepage_unittest.dll"
   "C:\Projects\Tests\bin\AddCart_unittest.dll"
```

xUnit.net

The xUnit.net framework provides two command-line runners: `xunit.console.exe` and `xunit.console.x86.exe`. You use the following syntax:

`xunit.console <assemblyFile> [configFile] [assemblyFile [configFile]...] [options] [reporter] [resultFormat filename [...]]`

xUnit.net accepts `.json` and `.xml` file extensions as configuration files (`configFile`).

You must call the appropriate executable with a list of DLLs, which are the test file names that you want to run. The following sample code runs two test files:

```
C:\xunit\xunit.console.exe
   "C:\Projects\Tests\bin\TestHomepage_unittest.dll"
   "C:\Projects\Tests\bin\AddCart_unittest.dll"
```

Java TestNG

The TestNG framework requires `testng.jar` libraries with a classpath (`-cp`) option and the `java.exe` application. In the `-cp` option, you must list all the library classes that you need to run your project. You use the following syntax:

`java -cp "<Path_to_testngjar>/testng.jar:<Path_to_Test_Classes>" org.testng.TestNG <Path_to_Test_xml>`

The following sample code runs an XML test file:

```
C:\Program Files\Java\jdk-12.0.1\bin\java.exe -cp
   ".\libs\: C:\Program Files\jbdevstudio4\studio\plugins\*"
   org.testng.TestNG testng.xml
```

Java JUnit

The JUnit framework has several versions and each version has its own command to execute tests. In the `-cp` option, you must list all the library classes that you need to run your project.

JUnit version 5.*x* uses the following syntax:

`java -jar junit-platform-console-standalone-<version>.jar --class-path <Path_to_Compiled_Test_Classes> --scan-class-path`

JUnit version 4.*x* uses the following syntax:

`java -cp .\libs\:<Path_to_Junitjar>\junit.jar org.junit.runner.JUnitCore [test class name]`

JUnit version 3.*x* uses the following syntax:

`java -cp .\libs\:<Path_to_Junitjar>\junit.jar junit.textui.TestRunner [test class name]`

The following sample code runs a test class:

```
C:\Program Files\Java\jdk-12.0.1\bin\java -cp
   C\java\libs\:C:\junit\junit.jar org.junit.runner.JUnitCore
   C:\project\test.class
```

Python unittest and PyUnit

Python provides built-in unit test modules (`-m`): Python unittest and PyUnit, depending on the version of Python you are using. These frameworks use the following syntax:

`python -m unittest [options] [tests]`

In this syntax, the `[tests]` can be a list of any number of test modules, classes, and test methods. The following command displays the unittest help in Python:

```
python -m unittest -h
```

The following sample code runs a test file named `tests.py` in the unittest module:

```
C:\Python\Python37-32\python.exe -m unittest
   C:\SampleProjects\POMProjectDemo\Tests\tests.py
```

Ruby RSpec

The RSpec framework provides unit testing libraries for Ruby code. This framework uses the following syntax:

```
<Path_to_RSpec>\rspec.bat [options] [files or directories]
```

The following sample code runs a test library:

```
C:\Ruby26-x64\bin\rspec.bat  -I C:\Ruby26-x64\Project\lib\
   C:\Ruby26-x64\Project\spec\calculator_spec.rb
```

JavaScript Jest

Jest a JavaScript library for creating and running tests on JavaScript code. This framework uses the following syntax:

```
<Path_to_Jest>\jest.js [--config=<pathToConfigFile>] [TestPathPattern]
```

The following sample code runs a test library:

```
C:\Users\admin\AppData\Roaming\npm\jest.cmd"
   --config=C:\Users\admin\AppData\Roaming\npm\jest.config.js
   C:/Users/admin/AppData/Roaming/npm/sum.test.js
```

Uploading files to OpenText DAST

To run a scan in the command-line interface (CLI) or create a macro using the API, you must upload all scripts and their dependencies to the machine where OpenText DAST is installed.

Using the CLI

To run a scan from the CLI, you must manually copy the files to the machine where OpenText DAST is installed.

Using the API

The OpenText DAST REST API provides the following endpoints for deploying these files:

- `POST /configuration/selenium/folder` – Upload and unzip ZIP file(s)
- `GET /configuration/selenium/folder` – Get a list of ZIP files that are already uploaded
- `GET /configuration/selenium/file/{foldername}` – Get a list of files that are contained in the ZIP file
- `DELETE /configuration/selenium/folder/{foldername}` – Delete the ZIP file

For details about using these endpoints, see the specific endpoint methods in the Swagger UI. For more information, see "Accessing the OpenText DAST API Swagger UI" on page 316.

Using the Selenium command

After creating and testing the Selenium command, you can use it to run a scan using WI.exe or create a macro using the API.

Important! When you conduct a scan using a Selenium command, a log directory is created in one of the following locations:

`C:\Users\<UserName>\AppData\Local\Temp\`

`C:\Windows\Temp` (when the OpenText DAST REST API is running under the system user)

If you end the geckodriver.exe or chromedriver.exe process while the scan is running, these temporary files will not be removed. You must manually remove these files.

Running a scan using WI.exe

For the command-line interface (CLI), WI.exe includes a `-selenium_workflow` parameter that accepts an XML object called `ArrayOfSeleniumCommand` as a file or a string.

Important! If you run a command as a string rather than a file, and the command contains the double-quotation mark character ("), then the character must be escaped with the backslash character (\) when you save it in the `<Command>` tag. For example, if the command includes spaces in the path, and you use double-quotation marks to pass the path in the Command, then the quotation marks must be escaped as shown here:

```
<Command>\"C:\Program Files\nunit\nunit3-console.ex\"
   C:\Projects\Tests\bin\TestHomepage_unittest.dll
   \"C:\Projects\Tests Main\bin\AddCart_unittest.dll\"</Command>
```

You place the Selenium command you created previously in the Command tag in the following syntax. For more information, see "Creating a Selenium command" on page 336.

```
<ArrayOfSeleniumCommand>
    <SeleniumCommand>
        <Command>"Commands"</Command>
        <AllowedHosts>
            <string>http://hostname/</string>
        </AllowedHosts>
```

```
        <WorkingDirectory>C:\pathtoprojectfolder\</WorkingDirectory>
    </SeleniumCommand>
    <SeleniumCommand>
        ...
    </SeleniumCommand>
    ...
</ArrayOfSeleniumCommand>
```

To pass the command as a file, use the following syntax:

```
-selenium_workflow "@PathToFile"
```

The following sample code pass a file named `wd_firefox.txt` as the command:

```
-selenium_workflow "@D:\tmp\selenium_wd\wd_firefox.txt"
```

For more information, see "Using wi.exe" on page 284.

Creating a macro using the API

To create a macro using the API, use the following endpoint:

```
POST /configuration/selenium/macro
```

The following sample code adds a macro using cURL:

```
curl -X POST --header "Content-Type: application/json" -d
   "{\"VerifyMacro\":true,\"name\": \"test\",\"command\":
   \"D:\\tmp\\selenium_wd\\bin\\net35\\nunit3-console.exe
   \\\"D:\\tmp\\selenium_wd\\selenium_c_sharp-master\\Selenium\\
   bin\\Debug\\Selenium.dll\\\"\",\"allowedHosts\":
   [\"http://zero.webappsecurity.com\"]}"
   http://localhost:8083/webinspect/configuration/selenium/macro
```

The following sample code starts a scan using cURL:

```
curl.exe -X POST --header "Content-Type: application/json"
   --header "Accept: application/json" -d "{\"settingsName\":
   \"Default\", \"overrides\": { \"startOption\": \"macro\",
   \"workflowMacros\": [\"test \"],\"AllowedHosts\":[\"\\*\"] ,
   \"crawlAuditMode\": \"auditOnly\"  } }"
   http://localhost:8083/webinspect/scanner/scans
```

Complete usage information and sample code are included in the Swagger UI, and objects are similar to those described in "Running a scan using WI.exe" on the previous page. For more information, see "Using the Swagger UI" on page 317.

The `WorkingDirectory` and `AllowedHosts` arguments are optional. In some cases, AllowedHosts can be determined automatically. However, OpenText recommends that you set AllowedHosts for each macro.

In some cases, you must set the Working Directory path, which is the "current working directory," for the `WorkingDirectory` argument.

About the Burp API extension

The Burp Suite is a toolkit for performing security testing of web applications. OpenText DAST includes a Burp extension that allows Burp Suite users to connect OpenText DAST to Burp via the OpenText DAST API.

Benefits of using the Burp API extension

Connecting OpenText DAST to Burp provides the following benefits:

- Create Burp issues with vulnerabilities from an OpenText DAST scan
  - Request vulnerabilities detected in a currently running or completed scan
  - Request vulnerabilities based on a specified criteria, such as Severity

  Note: OpenText DAST check IDs and names do not map to Burp issue IDs and names.
- Select sessions in Burp and send to OpenText DAST

  Note: Sessions could be selected for the following reasons:
  - Locations that need to be added to OpenText DAST's crawl in a running scan
  - New vulnerabilities that need to be added to a running scan
  - New vulnerabilities that need to be added to a completed scan
- Get Scan Information from OpenText DAST
  - Get status of a specific scan
  - Get a list of scans available in the currently connected OpenText DAST database
  - Get a list of scans based on scan status (Running/Complete)

Supported versions

The OpenText DAST Burp API extension is compatible with the new Burp Extension API.

See Also

"OpenText DAST REST API" on page 313

"Using the Burp API extension" on the next page

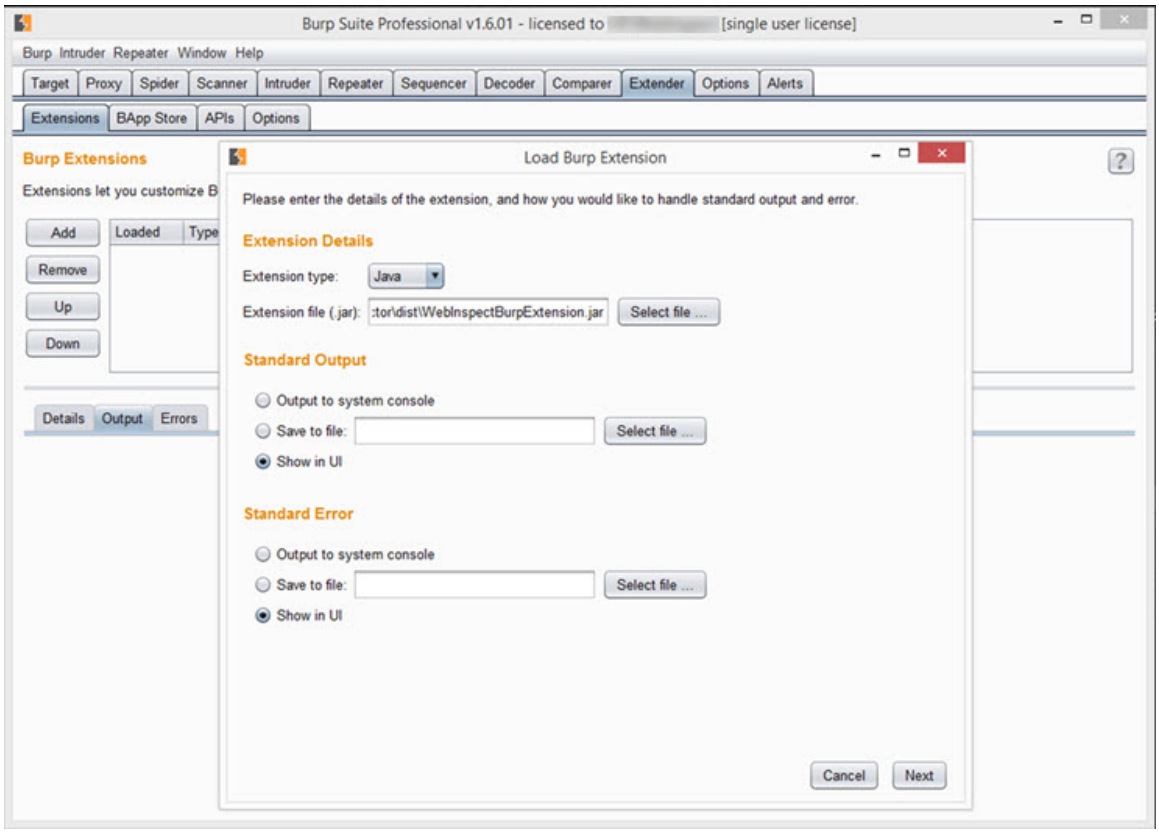
Using the Burp API extension

This topic describes how to set up and use the WebInspect Burp extension.

Loading the Burp extension

Perform the following steps in Burp to load the WebInspect Burp extension:

1. On the **Extender** tab, select **Extensions** and click **Add**.
   The Load Burp Extension window appears.



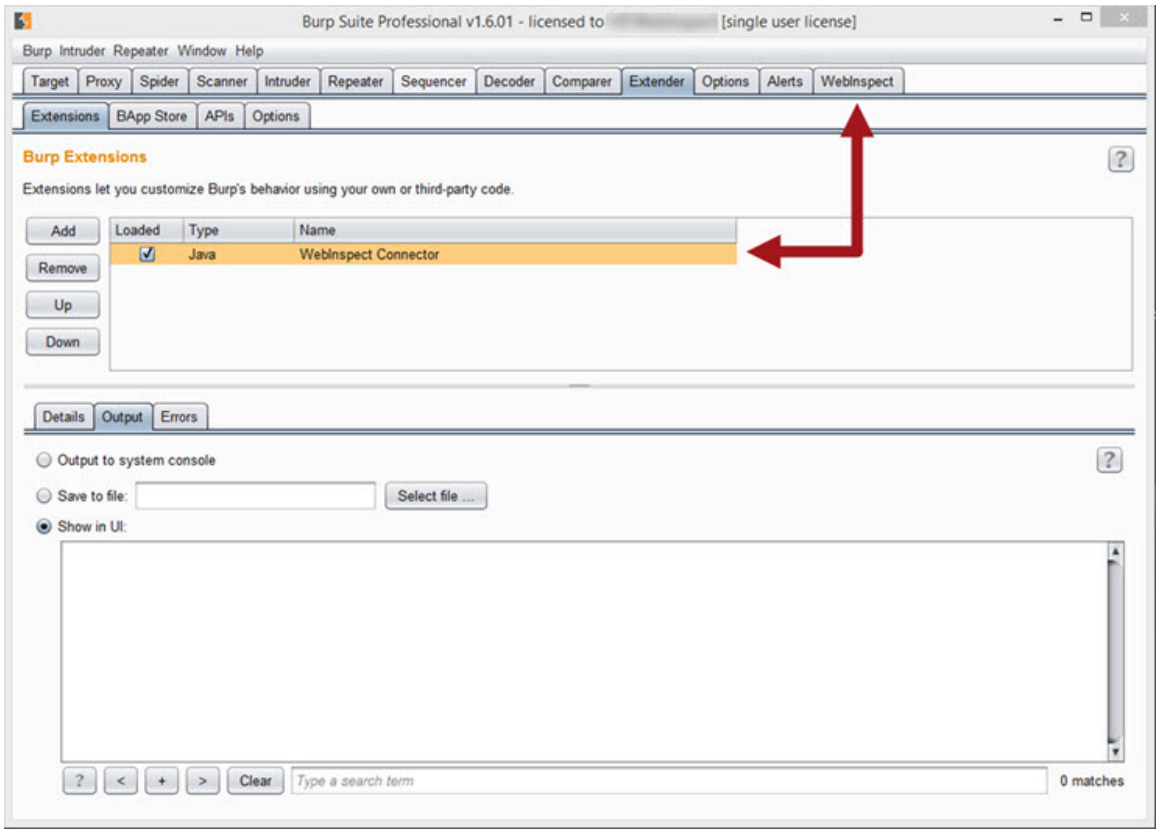
2. In the **Extension file (.jar)** field, click **Select file** and navigate to the WebInspectBurpExtension.jar file.

   Tip: The WebInspectBurpExtension.jar file can be found in the Extensions directory in the OpenText DAST installation location. The default location is:

   C:\Program Files\Fortify\Fortify WebInspect\Extensions

3. Ensure that the **Show in UI** option is selected under the **Standard Output** and **Standard Error** sections.
4. Click **Next**.

WebInspect Connector appears in the list of Burp Extensions and a tab labeled "WebInspect" is added to the Burp user interface. If you do not see the WebInspect tab, then the Burp extension did not load correctly. In this case, look in the Output and Errors tabs for information that may help you to troubleshoot the issue.

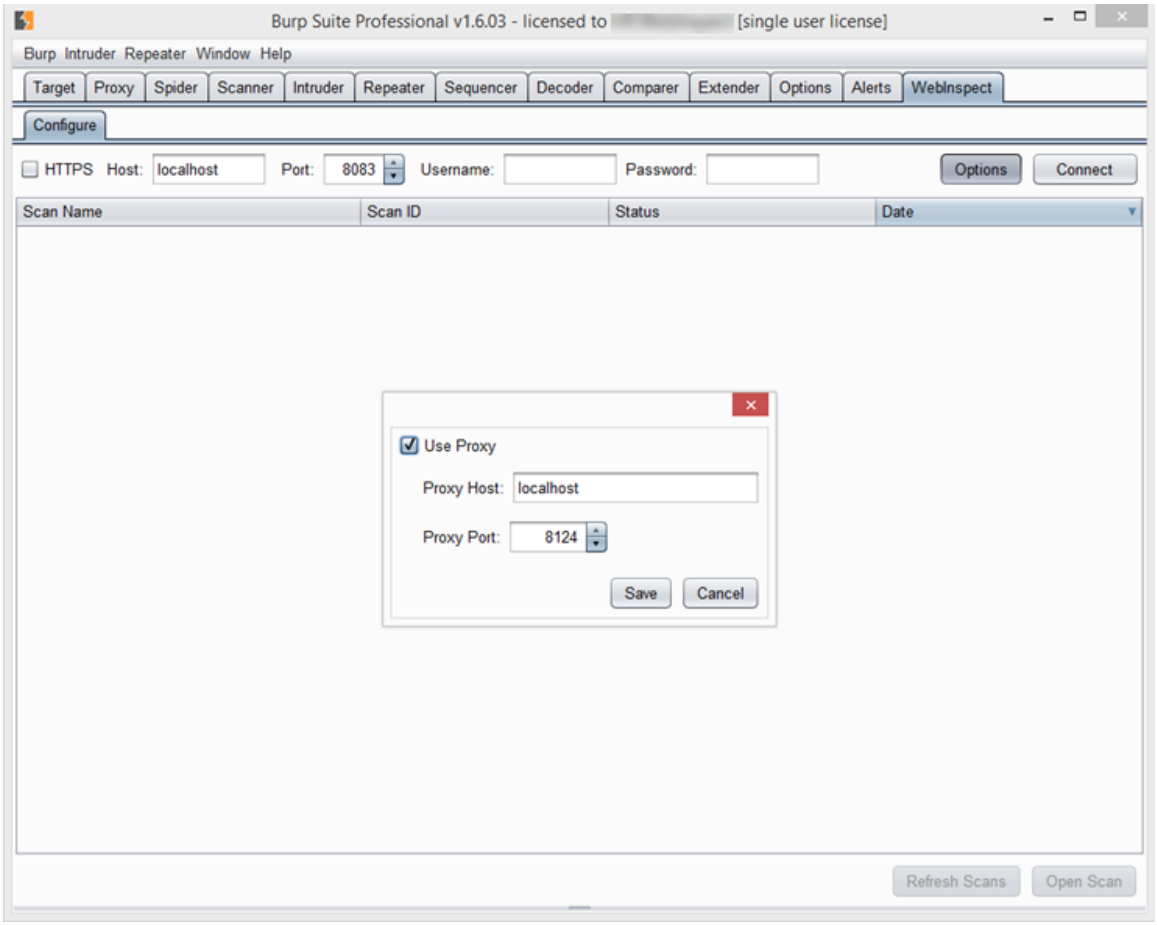


Connecting to OpenText DAST

Perform the following steps in Burp to connect to OpenText DAST:

1. Ensure that the OpenText DAST API service is running. For more information, see "OpenText DAST Monitor" on page 93.
2. On the **WebInspect > Configure** tab, do the following:
   a. If the API requires HTTPS authentication, select the **HTTPS** check box.
   b. Type the **Host** name and **Port** number for the OpenText DAST API service.
   c. If the API is configured to require authentication, type the **Username** and **Password**.
   d. Click **Options** to configure proxy settings for the API HTTP requests.

A proxy settings window appears.

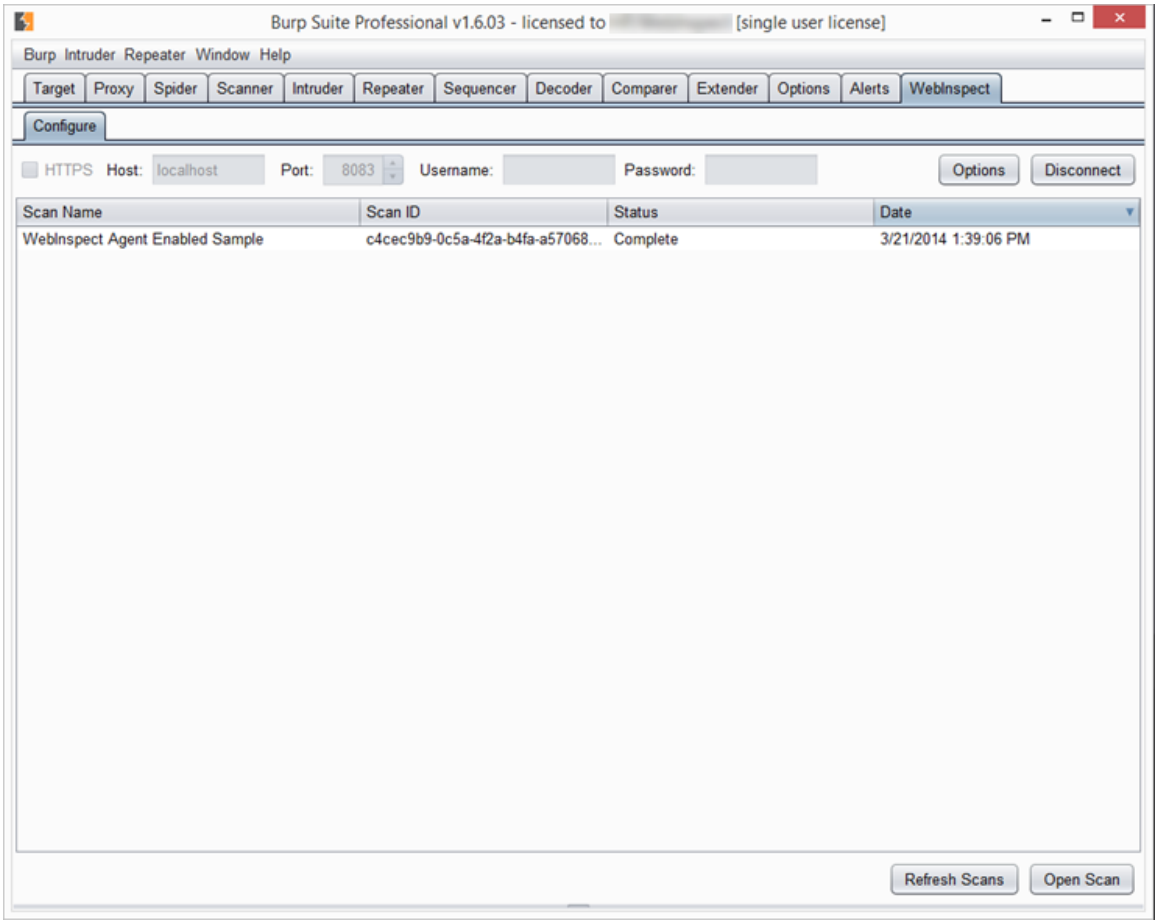


e. Select the **Use Proxy** checkbox, and type the **Proxy Host** name and the **Proxy Port** number.

f. Click **Save**.

3. Click **Connect**.

A list of OpenText DAST scans should appear in the WebInspect tab.



Refreshing the list of scans

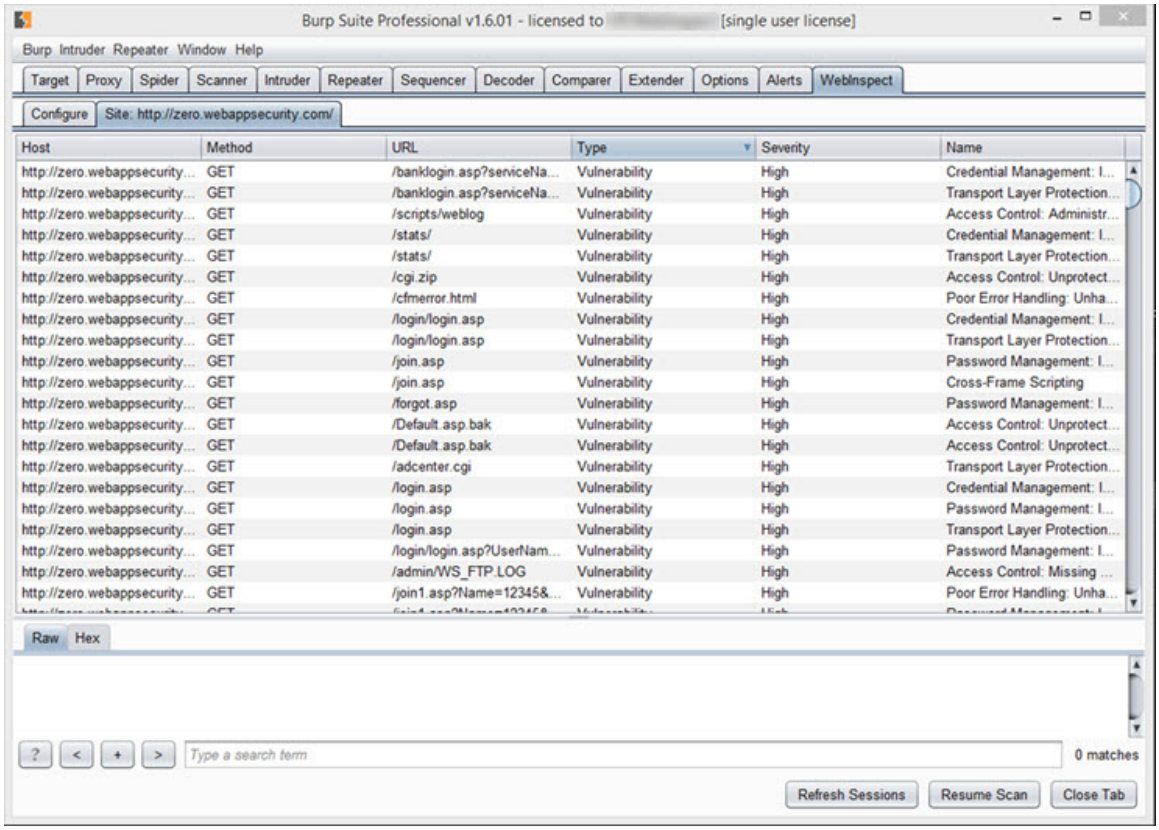
To update the list of OpenText DAST scans, click **Refresh Scans**.

Working with a scan in Burp

Perform the following steps in Burp to work with an OpenText DAST scan:

1. Do one of the following to open a scan:
   - Double-click on a scan in the list.
   - Select a scan in the list and click **Open Scan**.

The scan opens in a new tab under the WebInspect tab, with Crawl sessions and Vulnerable sessions listed. The list of sessions is automatically sorted by Type with Vulnerabilities first followed by Crawl sessions.



2. To re-sort on a sorted column in reverse order, click the column heading. To sort the list using different sort criteria, click the heading of the column you want to sort by. The following table describes some sort scenarios:

If you...	Then Sort By...
Have multiple hosts in the scan and want to group sessions by hosts	Host
Want to see all sessions that used a specific method	Method and scroll to the specific method you want
Want to see all sessions affecting a specific page in your website	URL and scroll to the specific page you want
Want to select all sessions with Critical and High severities and send them to a Burp tool	Severity and scroll to the sessions with Critical and High severities

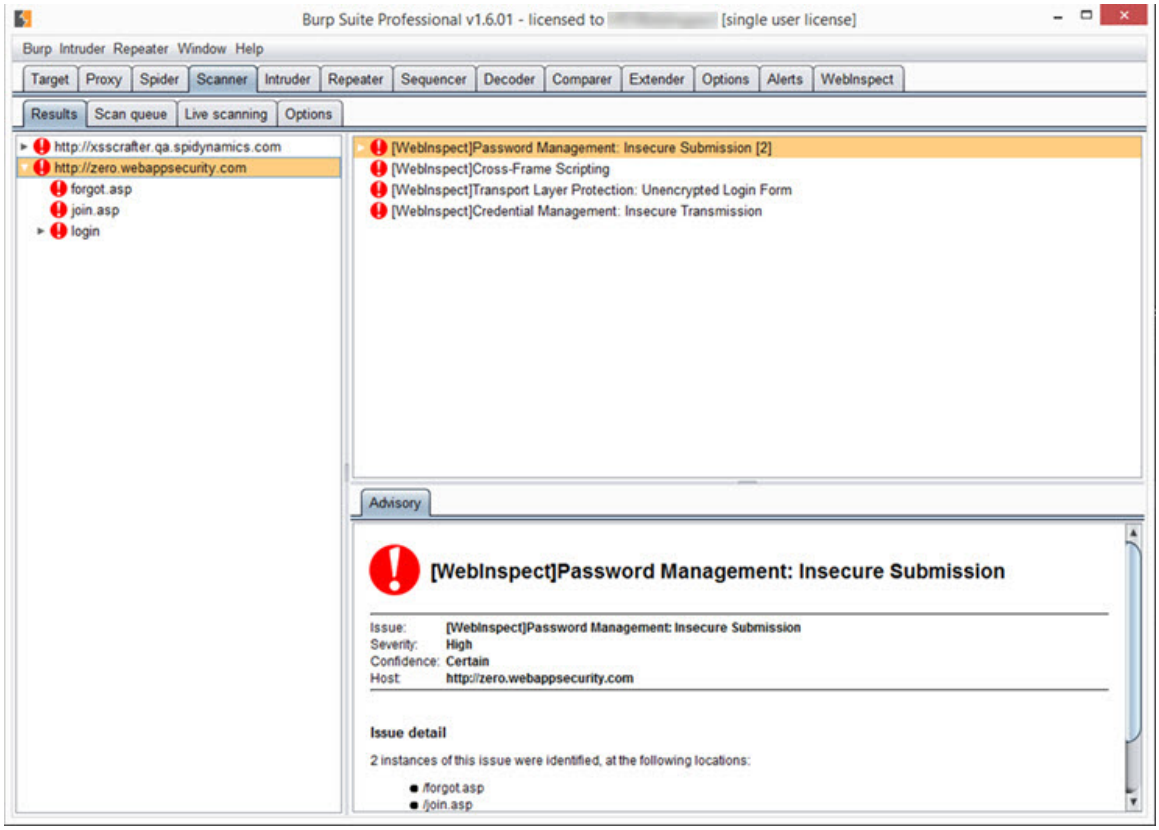
If you...	Then Sort By...
Want to select all sessions with the same check name	Name and scroll to the specific check name you want

3. To update the list of sessions—such as when Burp is connected to a scan that is still running—click **Refresh Sessions**.
4. To view the request for a session, click the session in the list.

   The session request information appears at the bottom of the window. Click the request to see the response.
5. To send one or more sessions to a Burp tool for further analysis, select the session (s), right-click and select the appropriate "**Send To**" option.

   Note: Current options are Send To Spider, Send To Intruder, and Send To Repeater. For more information about Burp tools, see the Burp Suite documentation.
6. To create an issue for a Vulnerable session and add it to the Scanner tab in Burp, right-click on the session and select **Create Issue**.

   The issue is populated with report data from OpenText DAST and the issue name is tagged with [WebInspect] to indicate that the issue was added from an external resource.



Note: The Create Issue option is only available in the Burp Professional Edition and is not available for Crawl sessions.

7. To continue a stopped scan, click **Resume Scan**.
8. To close the OpenText DAST scan, click **Close Tab**.

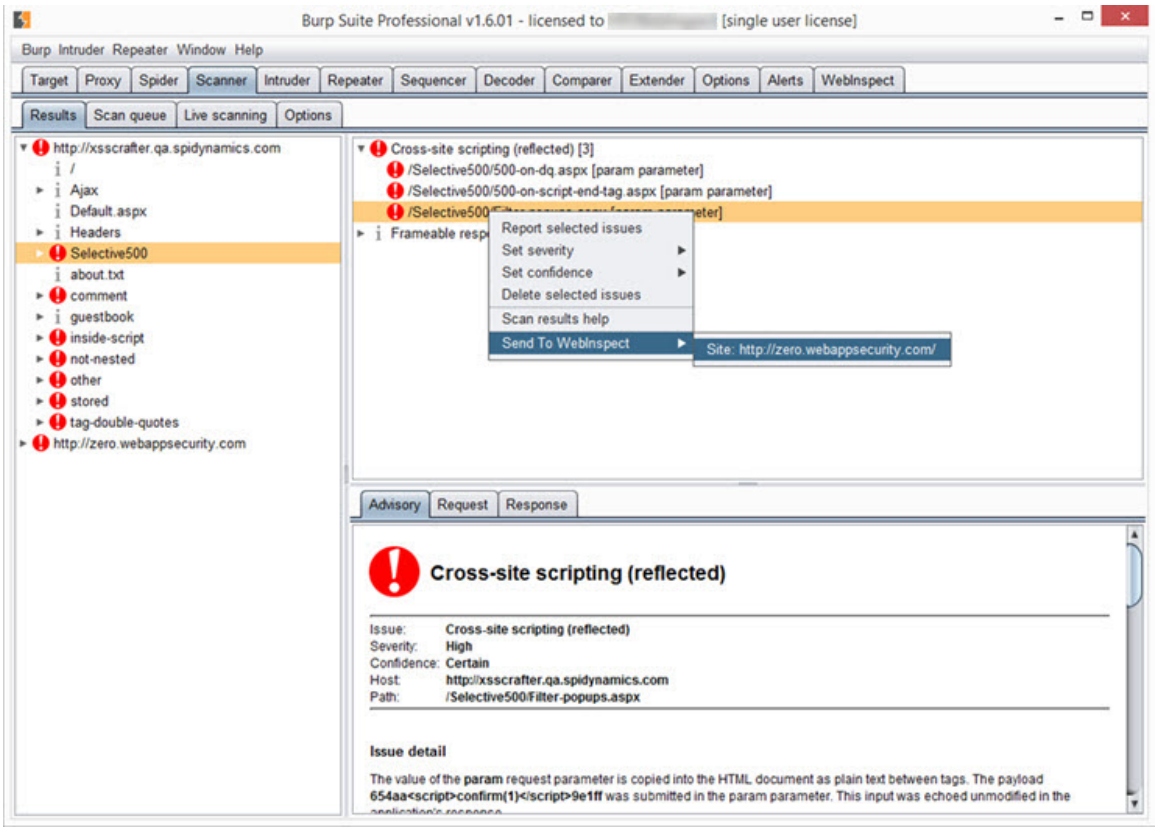
Sending items from Burp to OpenText DAST

Perform the following steps in Burp to send requests/responses and issues to OpenText DAST to be crawled:

1. Ensure that the desired OpenText DAST scan is open in the **WebInspect** tab.

   Tip: The Send To WebInspect option will not be available in the context menu if an OpenText DAST scan is not open in Burp.

2. Click the **Scanner** tab and then the **Results** tab.
3. To send a request/response to OpenText DAST to be crawled, right click the request and select **Send To WebInspect > [scan name]**.

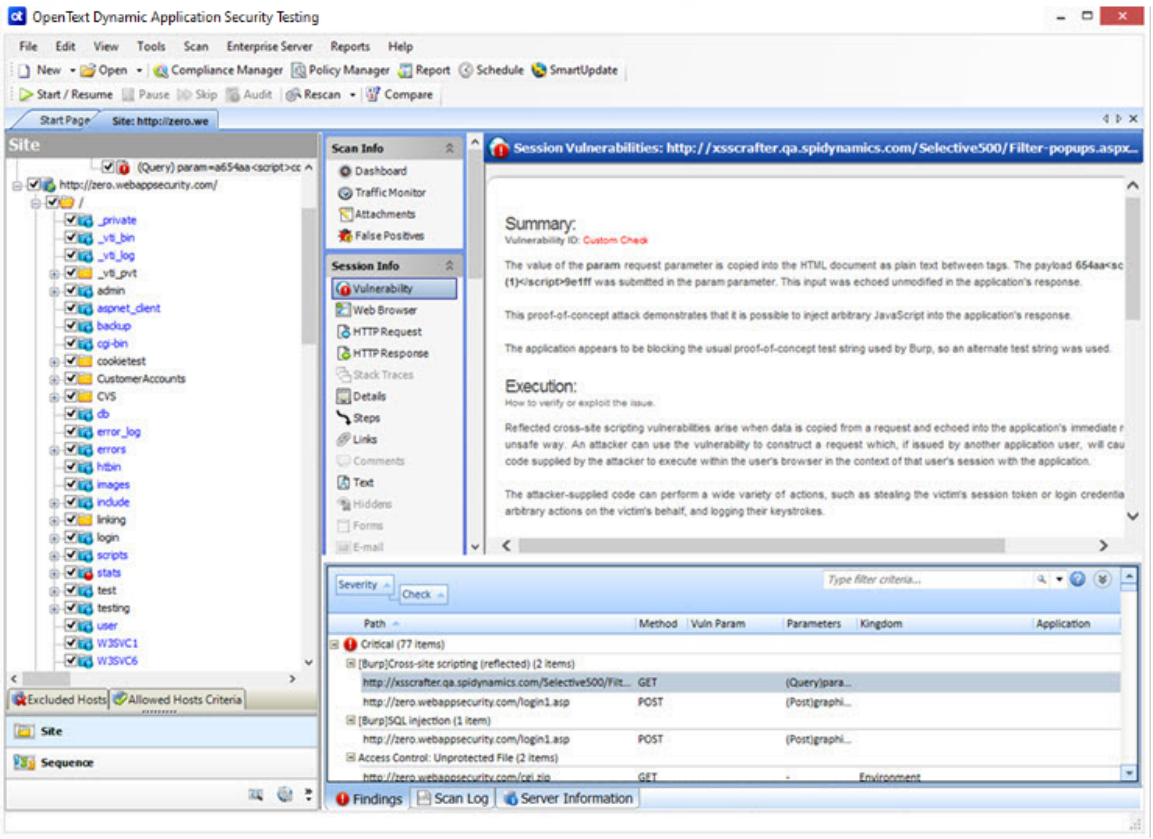


OpenText DAST creates a session for the request that is ready to be crawled. You can return to the scan in the **WebInspect** tab and click **Resume Scan** to crawl the session.

Note: Scan settings for the open scan apply to the session being sent. This may affect what OpenText DAST does with the session. For instance, if the open scan is for Host A and you send a session from Host B, but Host B is not in the Allowed Hosts list for the open scan, the session will be excluded and will not be crawled.

4. To send an issue to OpenText DAST as a manual finding, right click the issue and select **Send To WebInspect > [scan name]**.

   The issue is populated with report data from Burp and the issue name is tagged with [Burp] to indicate that the issue was added from an external resource.



See also

"About the Burp API extension" on page 342

"OpenText DAST REST API" on page 313

"OpenText DAST Monitor" on page 93

About the WebInspect SDK

The WebInspect Software Development Kit (SDK) is a Visual Studio extension that enables software developers to create an audit extension to test for a specific vulnerability in a session response.

Caution! OpenText recommends that the WebInspect SDK be used only by qualified software developers who have expertise in developing code using Visual Studio.

Audit extensions / custom agents

The WebInspect SDK provides the developer with entry points into the OpenText DAST code. When OpenText DAST creates a request/response pair, the developer can examine the response and create an audit extension that will flag a vulnerability. After the extension has been created, the developer sends it to the local copy of SecureBase, the OpenText DAST database of adaptive agents and vulnerability checks, where it is stored as a custom agent. The custom agent is assigned a Globally Unique Identifier (GUID) and becomes available for use in policies in the Policy Manager for an OpenText DAST product.

Note: Custom agents will not be overwritten by SecureBase updates.

When inspecting the scan results, you can perform the same actions—such as Copy URL and Review Vulnerability—on a vulnerability discovered by a custom agent as you can a vulnerability discovered by a standard check. For more information, see "Inspecting the results" on page 239.

SDK functionality

The SDK provides developers with the functionality to:

- Inspect sessions generated by the OpenText DAST crawler and auditor
- Inject values into parameters (parameter and sub-parameter fuzzing)
- Queue a URL for crawling (for the OpenText DAST crawler to crawl)
- Flag a vulnerability
- Send a raw HTTP request through the OpenText DAST requestor
- Request and response parsing via ParseLib
- Log events and errors

Installation recommendation

The WebInspect SDK does not need to be installed on the same machine as an OpenText DAST product. In most cases, it will be installed on the software developer's development machine. However, if you are developing new extensions that will require debugging, OpenText recommends that you install OpenText DAST on the development machine where you will be creating the extension. Doing so will allow you to test your extension locally. For existing extensions that do not require debugging, you do not need to install OpenText DAST locally.

Refer to the *OpenText™ Application Security Software System Requirements* document for minimum requirements for installing and using the WebInspect SDK.

Installing the WebInspect SDK

To use the WebInspect SDK, the developer must install a Visual Studio extension file named WebInspectSDK.vsix.

During installation of OpenText DAST, a copy of the WebInspectSDK.vsix file is installed in the Extensions directory in the OpenText DAST installation location. The default location is:

- C:\Program Files\Fortify\Fortify WebInspect\Extensions

To install the local copy where OpenText DAST is installed on the developer's machine:

1. Navigate to the **Extensions** folder and double-click the **WebInspectSDK.vsix** file.
   The VSIX Installer is launched.
2. When prompted, select the Visual Studio product(s) to which you want to install the extension and click **Install**.
   The WebInspect Audit Extension project template is created in Visual Studio. Continue with "Verifying the installation" below.

To install the local copy where OpenText DAST is NOT installed on the developer's machine:

1. Navigate to the **Extensions** folder and copy the **WebInspectSDK.vsix** file to portable media, such as a USB drive.
2. Insert the drive into the development box that has Visual Studio 2013 installed, as well as the related required software and hardware.
3. Navigate to the USB drive and double-click the **WebInspectSDK.vsix** file.
   The VSIX Installer is launched.
4. When prompted, select the Visual Studio product(s) to which you want to install the extension and click **Install**.
   The WebInspect Audit Extension project template is created in Visual Studio. Continue with "Verifying the installation" below.

Verifying the installation

To verify that the extension was successfully installed:

1. In Visual Studio, select **Tools > Extensions and Updates**.
2. Scroll down the list of extensions.
   If you see **WebInspect SDK** in the list, the extension was installed successfully.

After installation

After installing and configuring the WebInspect SDK, the developer can create a new WebInspect Audit Extension project in Visual Studio. In this project, the developer will create an audit extension, debug and test the extension, and publish the extension to SecureBase as a custom agent. For information about using the WebInspect Audit Extension project template, refer to the WebInspect SDK documentation in Visual Studio.

After the developer has sent the custom agent to SecureBase, the agent can be selected in policies in the Policy Manager. See the Policy Manager documentation for more information.

Add page or directory

If you use manual inspection or other security analysis tools to detect resources that OpenText DAST did not discover, you can add these locations manually and assign a vulnerability to them. Incorporating the data into an OpenText DAST scan enables you to report and track vulnerabilities using OpenText DAST features.

Note: When creating additions to the data hierarchy, you must manually add resources in a logical sequence. For example, to create a subdirectory and page, you must create the subdirectory before creating the page.

1. Replace the default name of the page or directory with the name of the resource to be added.
2. If necessary, edit the HTTP request and response. Do not change the request path.
3. You can send a request to the resource and record the response in the session data. This will also verify the existence of the resource that was not discovered by OpenText DAST:
   a. Click **HTTP Editor** to open the HTTP Editor.
   b. If necessary, modify the request.
   c. Click 

.
   d. Close the HTTP Editor.
   e. When prompted to use the modified request and response, select **Yes**.
4. (Optional) To delete all request and response modifications, click **Reset**.
5. When finished, click **OK**.

Add variation

If you use manual inspection or other security analysis tools to detect resources that OpenText DAST did not discover, you can add these locations manually and assign a vulnerability to them. Incorporating the data into an OpenText DAST scan enables you to report and track vulnerabilities using OpenText DAST features.

A variation is a subnode of a location that lists particular attributes for that location. For example, the login.asp location might have the variation:

(Post) uid=12345&Password=foo&Submit=Login

Variations, like any other location, can have vulnerabilities attached to them, as well as subnodes.

1. In the **Name** box, replace the default "attribute=value" with the actual parameters to be sent (for example, uid=9999&Password=kungfoo&Submit=Login.
2. Select either **Post** or **Query**.
3. If necessary, edit the HTTP request and response. Do not change the request path.
4. You can send a request to the resource and record the response in the session data. This will also verify the existence of the resource that was not discovered by OpenText DAST:
   a. Click **HTTP Editor** to open the HTTP Editor.
   b. If necessary, modify the request.
   c. Click 

.
   d. Close the HTTP Editor.
   e. When prompted to use the modified request and response, select **Yes**.
5. (Optional) To delete all request and response modifications, click **Reset**.
6. When finished, click **OK**.

Creating an exclusion

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, Post Parameter, or Response Header 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 OpenText's regular expression extensions you specify in the **Match String** box. For more information, see "Regex extensions" on page 311.
   - 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 treated as AND conditions.
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 **Other Exclusion/Rejection Criteria** list, select either **Reject**, **Exclude**, or both.

Note: You cannot reject Response, Response Header, and Status Code Target types during a scan. You can only exclude these Target types.

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, OpenText DAST 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]*/

Internet Protocol version 6

OpenText DAST supports Internet Protocol version 6 (IPv6) addresses in website and web service scans. When you specify the Start URL, you must enclose the IPv6 address in brackets. For example:

- http://[::1]
  OpenText DAST scans "localhost."

- http://[fe80::20c:29ff:fe32:bae1]??/subfolder/??

  OpenText DAST scans the host at the specified address starting in the "subfolder" directory.
- http://[fe80::20c:29ff:fe32:bae1]??:8080/subfolder/??

  OpenText DAST scans a server running on port 8080 starting in "subfolder."

Chapter 6: Default scan settings

This chapter describes the default scan settings. Use default settings to establish scanning parameters for your scan actions. OpenText DAST uses these options unless you specify alternatives while initiating a scan (using the options available through the Scan Wizard or by accessing Current Settings).

See also

"Crawl settings" on page 408

"Audit settings" on page 421

Scan settings: Method

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Method**.

Scan mode

The Scan Mode options are described in the following table.

Option	Description
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 OpenText DAST 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 Crawl and Audit Mode option called Simultaneously. For more information, see "Crawl and audit mode" on the next page.
Audit Only	OpenText DAST 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 enables you to navigate manually to whatever

Option	Description
(Not available for Guided Scan)	sections of your application you choose to visit. 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. After 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

The Crawl and Audit Mode options are described in the following table.

Option	Description
Simultaneously	As OpenText DAST 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, OpenText DAST crawls the entire site, mapping the site's hierarchical data structure, and then conducts a sequential audit, beginning at the site's root.

Crawl and audit details

The Crawl and Audit Details options are described in the following table.

Option	Description
Include search probes (send search attacks)	If you select this option, OpenText DAST will send requests for files and directories that might or might not exist on the server, even if those files are not found by crawling the site. This option is selected by default only when the Scan Mode is set to Crawl & Audit. The option is cleared(unchecked) by default when the Scan Mode is set to Crawl Only or Audit Only.
Crawl links on File	If you select this option, OpenText DAST will look for and crawl

Option	Description
Not Found responses	links on responses that are marked as “file not found.” This option is selected by default when the Scan Mode is set to Crawl Only or Crawl & Audit. The option is not available when the Scan Mode is set to Audit Only.

Navigation

The Navigation options are described in the following table.

Option	Description
Auto-fill Web forms during crawl	If you select this option, OpenText DAST 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 select the **Edit** button  (to modify the currently selected file) or the **Create** button  (to create a web form file). **Caution!** Do not rely on this feature for authentication. If the crawler and the auditor are configured to share state, and if OpenText DAST 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 OpenText DAST will not be able to log in again and the auditing will be unauthenticated. To prevent OpenText DAST 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, OpenText DAST pauses the scan when it encounters an HTTP or JavaScript form and displays a window that enables you to enter values for input controls within the form. However, if you also select **Only prompt for tagged inputs**, OpenText DAST 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.

Option	Description
	For more information about configuring an interactive scan, see "Interactive scans" on page 194.

SSL/TLS protocols

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols provide secure HTTP (HTTPS) connections for Internet transactions between web browsers and web servers. SSL/TLS protocols enable server authentication, client authentication, data encryption, and data integrity for web applications.

Note: If **Use OpenSSL Engine** is selected in Application Settings, the SSL/TLS Protocols options are disabled. You cannot select individual protocols. For more information, see "Application settings: General" on page 431.

Select the SSL/TLS protocol(s) used by your web server. The following options are available:

- Use SSL 2.0
- Use SSL 3.0
- Use TLS 1.0
- Use TLS 1.1
- Use TLS 1.2

If you do not configure the SSL/TLS protocol to match your web server, OpenText DAST will still connect to the site, though there may be a performance impact.

For example, if the setting in OpenText DAST is configured to Use SSL 3.0 only, but the web server is configured to accept TLS 1.2 connections only, OpenText DAST will first try to connect with SSL 3.0, but will fail. OpenText DAST will then implement each protocol until it discovers that TLS 1.2 is supported. The connection will then succeed, although more time will have been spent in the effort. The correct setting (Use TLS 1.2) in OpenText DAST would have succeeded on the first try.

Scan settings: General

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **General**.

Scan details

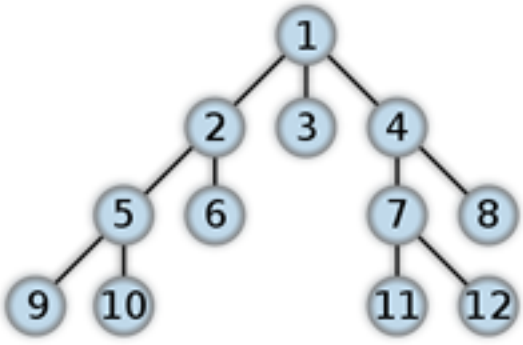
The Scan Details options are described in the following table.

Option	Description
Enable Path Truncation	Path truncation attacks are requests for known directories without file names. This may cause directory listings to be displayed. OpenText DAST 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.
Case-sensitive request and response handling	Select this option if the server at the target site is case-sensitive to URLs.
Recalculate correlation data	This option is used only for comparing scans. The setting should be changed only upon the advice of Customer Support personnel.
Compress response data	If you select this option, OpenText DAST saves disk space by storing each HTTP response in a compressed format in the database.
Enable Traffic Monitor Logging	During a Basic Scan, OpenText DAST displays in the navigation pane only those sessions that reveal the hierarchical structure of the website plus those sessions in which a vulnerability was discovered. However, if you select the Traffic Monitor option, OpenText DAST adds the **Traffic Monitor** button to the Scan Info panel, allowing you to display and review every single HTTP request sent by OpenText DAST and the associated HTTP response received from the server.
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.

Option	Description
	Encrypted files cannot be compressed. Selecting this option will significantly increase the size of exported scans containing log files. **Note:** The Traffic Viewer tool does not support the encryption of traffic files. The **Encrypt Traffic Monitor File** option is reserved for use under special circumstances with legacy traffic files only.
Maximum crawl-audit recursion depth	When an attack reveals a vulnerability, OpenText DAST 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

By default, OpenText DAST uses breadth-first crawling, which 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.



You cannot change this crawling method in the user interface. However, the configurable Crawl Details options are described in the following table.

Option	Description
Enable keyword search audit	A keyword search, as its name implies, uses an attack engine that examines server responses and searches for certain text strings that typically indicate a vulnerability. Normally, this engine is not used during a crawl-only scan, but you can enable it by selecting this option.
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, OpenText DAST would never be able to finish the scan. This option compares page structure to determine the level of similarity, allowing OpenText DAST to identify and exclude processing of redundant resources. **Important!** Redundant page detection works in the crawl portion of the scan. If the audit introduces a session that would be redundant, the session will not be excluded from the scan. You can configure the following settings for redundant page detection: • **Page Similarity Threshold** – indicates how similar two pages must be to be considered redundant. Enter a percentage from 1 to 100, where 100 is an exact match. The default setting is 95 percent. • **Tag attributes to include** - identifies the tag attributes to include in the page structure. Typically, tag attributes and their values are dropped when determining structure. Identifying tag attributes in this field in a comma-separated list adds those attributes and their values in the page structure. By default, "`id,class`" tag attributes are included. **Tip:** Certain sites may be primarily composed of one type of tag, such as `<div>`. Including these attributes creates a more rigid page match. Excluding these attributes creates a less strict match.
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

Option	Description
	number of times a single URL will be crawled. 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.apsx?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). **Note:** This setting applies to both GET and POST parameters.
Limit maximum directory hit count to	This setting defines the maximum number of sub-directories and pages to be traversed within each directory during the crawl. This setting reduces the scope of the crawl and might be useful in reducing scan times for some sites, such as those consisting of a content management system (CMS). The default setting is 200.
Minimum folder depth	If you select **Limit maximum directory hit count to** (above), this setting defines the folder depth at which the maximum directory hit count will begin to apply. The default setting is 1.
Limit maximum link traversal sequence to	This option restricts the number of hyperlinks that can be sequentially accessed as OpenText DAST 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

Option	Description
	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. The default value is 15. 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.
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 a scan of a large site. **Note:** The limit set here does not directly correlate to the Crawled progress bar that is displayed during a scan. The maximum crawl count set here applies to links found by the Crawler during a crawl of the application. The Crawled progress bar includes all sessions (requests and responses) that are parsed for links during a crawl and audit, not just the links found by the Crawler during a crawl.
Limit maximum Web form submission to	Normally, when OpenText DAST 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 OpenText DAST will perform. The default value is 3.
Suppress Repeated Path Segments	Many sites have text that resembles relative paths that become unusable URLs after OpenText DAST parses them and appends them to the URL being crawled. These occurrences

Option	Description
	can result in a runaway scan if paths are continuously appended, such as `/foo/bar/foo/bar/`. This setting helps reduce such occurrences and is enabled by default. With the setting **enabled**, the options are: **1** – Detect a single sub-folder repeated anywhere in the URL and reject the URL if there is a match. For example, `/foo/baz/bar/foo/` will match because "`/foo/`" is repeated. The repeat does not have to occur adjacently. **2** – Detect two (or more) pairs of adjacent sub-folders and reject the URL if there is a match. For example, `/foo/bar/baz/foo/bar/` will match because "`/foo/bar/`" is repeated. **3** – Detect two (or more) sets of three adjacent sub-folders and reject the URL if there is a match. **4** – Detect two (or more) sets of four adjacent sub-folders and reject the URL if there is a match. **5** – Detect two (or more) sets of five adjacent sub-folders and reject the URL if there is a match. If the setting is **disabled**, repeating sub-folders are not detected and no URLs are rejected due to matches.

Scan settings: JavaScript

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **JavaScript**.

JavaScript settings

The JavaScript analyzer allows OpenText DAST to crawl links defined by JavaScript, and to create and audit any documents rendered by JavaScript.

Tip: To increase the speed at which OpenText DAST conducts a crawl while analyzing script, change your browser options so that images/pictures are not displayed.

Configure the settings as described in the following table.

Option	Description
Crawl links found from script execution	If you select this option, the crawler will follow dynamic links (i.e., links generated during JavaScript execution).
Verbose script parser debug logging	If you select this setting AND if the Application setting for logging level is set to Debug, OpenText DAST logs every method called on the DOM object. This can easily create several gigabytes of data for medium and large sites.
Log JavaScript errors	OpenText DAST logs JavaScript parsing errors from the script parsing engine.
Enable JS Framework UI Exclusions	With this option selected, the OpenText DAST JavaScript parser ignores common JQuery and Ext JS user interface components, such as a calendar control or a ribbon bar. These items are then excluded from JavaScript execution during the scan.
Enable Site-Wide Event Reduction	When this option is selected, the crawler and JavaScript engine recognize common functional areas that appear among different parts of the website, such as common menus or page footers. This eliminates the need to find within HTML content the dynamic links and forms that have already been crawled, resulting in quicker scans. This option is enabled by default and should not normally be disabled.
Capture WebSocket Events	WebSocket is an asynchronous protocol, which means that not every request requires a response. Most of the time when a request does not receive a response, WebSocket ends with a timeout that affects both scan time and the ability to discover new attack surface. To prevent adversely affecting scan quality, this option is disabled by default.
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.
SPA support	SPA support applies to single-page applications. When enabled, the DOM script engine finds JavaScript includes, frame and iframe includes, CSS file includes, and AJAX calls during the crawl, and then audits all traffic generated by those events. Options for SPA support are:

Option	Description
	• **Automatic** - If OpenText DAST detects a SPA framework, it automatically switches to SPA-support mode. • **Enabled** - Indicates that SPA frameworks are used in the target application. **Caution!** SPA support should be enabled for single-page applications only. Enabling SPA support to scan a non-SPA website will result in a slow scan. • **Disabled** - Indicates that SPA frameworks are not used in the target application. For more information, see "About single-page application scans" on page 201.

Scan settings: Requestor

A requestor is the software module that handles HTTP requests and responses.

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Requestor**.

Requestor performance

The Requestor Performance options are described in the following table.

Option	Description
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 OpenText DAST and is suitable for use when maintaining state is not a significant consideration. You also specify the maximum number of threads (up to 75).
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.

Option	Description
	When performing crawl and audit, you can specify the maximum number of threads that can be created for each requestor. The **Crawl requestor thread count** can be configured to send up to 25 concurrent HTTP requests before waiting for an HTTP response to the first request; the default setting is 5. The **Audit requestor thread count** can be set to a maximum of 50; the default setting is 10. Increasing the thread counts may increase the speed of a scan, but might also exhaust your system resources as well as those of the server you are scanning. **Note:** Depending on the capacity of the application being scanned, increasing thread counts may increase request failures due to increased load on the server, causing some responses to exceed the **Request timeout** setting. Request failures may reduce scan coverage because the responses that failed may have exposed additional attack surface or revealed vulnerabilities. If you notice increased request failures, you might reduce them by either increasing the **Request timeout** or reducing the **Crawl requestor thread count** and **Audit requestor thread count**. Also, depending on the nature of the application being scanned, increased crawl thread counts may reduce consistency between subsequent scans of the same site due to differences in crawl request ordering. By reducing the default **Crawl requestor thread count** setting to 1, consistency may be increased.

Requestor settings

The Requestor Settings options are described in the following table.

Option	Description
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.

Option	Description
Request retry count	Specify how many times OpenText DAST 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 OpenText DAST will wait for an HTTP response from the server. If this threshold is exceeded, OpenText DAST resubmits the request until reaching the retry count. If it then receives no response, OpenText DAST logs the timeout and issues the first HTTP request in the next attack series. The default value is 20 seconds. **Note:** The first time a timeout occurs, OpenText DAST will extend the timeout period to confirm that the server is unresponsive. If the server responds within the extended Request timeout period, then the extended period becomes the new Request timeout for the current scan.

Stop scan if loss of connectivity detected

There may be occasions during a scan when a web server fails or becomes too busy to respond in a timely manner. You can instruct OpenText DAST to terminate a scan by specifying a threshold for the number of timeouts.

The options are described in the following table.

Option	Description
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	Enter the total number of nonconsecutive timeouts permitted

Option	Description
"any host" retry failures to stop scan	from all hosts. The default value is 350.
If first request fails, stop scan	Selecting this option will force OpenText DAST to terminate the scan if the target server does not respond to OpenText DAST's first request.
Response codes to stop scan if received	Enter the HTTP status codes that, if received, will force OpenText DAST to terminate the scan. Use a comma to separate entries; use a hyphen to specify an inclusive range of codes.

Scan settings: Session Exclusions

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Session Exclusions**.

These settings apply to both the crawl and audit phases of an OpenText DAST vulnerability scan. To specify exclusions for only the crawl or only the audit, use the Crawl Settings: Session Exclusions or the Audit Settings: Session Exclusions.

Excluded or rejected file extensions

You can identify a file type and then specify whether you want to exclude or reject it.

- **Reject** - OpenText DAST will not request files of the type you specify.
- **Exclude** - OpenText DAST 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.

To add a file extension to reject or exclude:

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

OpenText DAST will not process files associated with the MIME type you specify. By default, image, audio, and video types are excluded.

To add a MIME Type to exclude:

1. Click **Add**.
   The Provide a Mime-type to Exclude window opens.
2. In the **Exclude Mime-type** box, enter a MIME type.
3. 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** - OpenText DAST 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, OpenText DAST will not examine the specified URL or host for links to other resources. During the audit portion of the scan, OpenText DAST 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.

Editing criteria

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**.

Adding criteria

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 Fortify'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  (or press Enter).
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 **Other Exclusion/Rejection Criteria** list, select either **Reject**, **Exclude**, or both.

Note: You cannot reject Response, Response Header, and Status Code Target types during a scan. You can only exclude these Target types.

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, OpenText DAST 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]*/

Scan settings: Allowed Hosts

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Allowed Hosts**.

Using the Allowed Host setting

Use the Allowed Host setting 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 OpenText DAST 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, OpenText DAST 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

Adding allowed domains

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**.

   Note: When specifying the URL, do not include the protocol designator (such as http:// or https://).

Editing or removing domains

To edit or remove an allowed domain:

1. Select a domain from the **Allowed Hosts** list.
2. Click **Edit** or **Remove**.

Scan settings: HTTP Parsing

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **HTTP Parsing**.

Options

The HTTP Parsing options are described in the following table.

Option	Description
HTTP Parameters Used for State	If your application uses URL rewriting or post data techniques to maintain state within a website, 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 OpenText DAST 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=slbhkelvbkl73dhj. In this case, "userid" is the parameter you would identify. **Note:** 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. OpenText DAST 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: /\([\w\d]+\)/
Enable CSRF	The Enable CSRF option should only be selected if the site you are scanning includes Cross-Site Request Forgery (CSRF) tokens as it

Option	Description
	adds overhead to the process. For more information, see "CSRF" on page 381.
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. Two default regular expressions identify two ASP.NET cookieless session IDs. The third regular expression matches jsessionid cookie.
Enable Response State Rules	If your application maintains client state with bearer tokens, select this option and create a rule that will identify the bearer token from the response and add it to the next request automatically. **Note:** The **Auto Response State Rules** option is enabled by default and provides several predefined rules for automatic detection of bearer tokens. You can enhance the automatic detection of bearer tokens by enabling response state rules and adding a rule as described in the following procedure. To add a rule: 1. After selecting the **Enable Response State Rules** check box, click **Add**. The Rule Search and Replace window appears. 2. In the **Rule Name** field, type a unique name for the rule. An example is `Bearer`. 3. Click **Add** next to the **Search in Response** field. The Search in Response dialog box opens in simple mode. **Note:** If you previously selected Regex Mode, the dialog box opens in Regex mode. 4. Do one of the following: • To create a rule in simple mode, type the text that contains the token in the **Rule** box. As you type, a regular expression is automatically generated in the **Regex View** box. **Tip:** Click ℹ to view a list of predefined tokens. • To use a predefined regular expression, select **Regex Mode** and select a regular expression statement from the **Regex** list. You can then edit the selected statement.

Option	Description
	5. Click **OK**. The regular expression is validated. You must correct any errors that are found before continuing. 6. Click **Add** next to the **Replace in Request** field. The Replace in Request dialog box opens in simple mode. **Note:** If you previously selected Regex Mode, the dialog box opens in Regex mode. 7. Do one of the following: • To create a rule in simple mode, type the text that contains the token in the **Rule** box. As you type, a regular expression is automatically generated in the **Regex View** box. **Tip:** Click ℹ to view a list of predefined tokens. • To use a predefined regular expression, select **Regex Mode** and select a regular expression statement from the **Regex** list. You can then edit the selected statement. 8. Click **OK**. The regular expression is validated. You must correct any errors that are found before continuing. 9. Click **OK** to close the Rule Search and Replace window. **Important!** To avoid regular expressions that could drain your system resources and affect scan performance, do not use the following text strings when constructing your regular expressions: • Any character with infinite numbers `".*"` or `".+"` • Positive lookahead `"(?=…)"` • Negative lookahead `"(?!...)"` • Positive lookbehind `"(?<=…)"` • Negative lookbehind `"(?<!...)"`
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;

Option	Description
	Ex. 3 – http://www.anysite.com?Master.asp?Page=13;Subpage=4 Ordinarily, OpenText DAST 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 website 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 webpages 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 language family (and implied character set) OpenText DAST should use.
Treat query parameter value as parameter name when only value is present	This setting defines how OpenText DAST interprets query parameters without values. For example: http://somehost?param If this checkbox is selected, OpenText DAST will interpret “param” to be a parameter named “param” with an empty value. If this checkbox is not selected, OpenText DAST will interpret “param” to be a nameless parameter with the value “param”. This setting can influence the way OpenText DAST calculates the hit count (see the "Limit maximum single URL hits to " on page 364 setting under Scan Settings: General). This setting is useful for scenarios in which a URL contains an anti-caching parameter. These often take the form of a numeric counter or timestamp. For example, the following

Option	Description
	parameters are numeric counters: • http://somehost?1234567 • http://somehost?1234568 In such cases, the value is changing for each request. If the value is treated as the parameter name, and the “Include parameters in hit count” setting is selected, the crawl count may inflate artificially, thus increasing the scan time. In these cases, clearing the “Treat query parameter value as parameter name when only value is present” checkbox will prevent these counters from contributing to the hit count and produce a more reasonable scan time.

CSRF

The Enable CSRF option should only be selected if the site you are scanning includes Cross-Site Request Forgery (CSRF) tokens as it adds overhead to the process.

About CSRF

Cross-Site Request Forgery (CSRF) is a malicious exploit of a website where unauthorized commands are transmitted from a user’s browser that the website trusts. CSRF exploits piggyback on the trust that a site has in a user’s browser; using the fact that the user has already been authenticated by the site and the chain of trust is still open.

Example:

A user visits a bank, is authenticated, and a cookie is placed on the user’s machine. After the user completes the banking transaction, he or she switches to another browser tab and continues a conversation on an enthusiast Web site devoted to the user’s hobby. On the site, someone has posted a message that includes an HTML image element. The HTML image element includes a request to the user’s bank to extract all of the cash from the account and deposit it into another account. Because the user has a cookie on his or her device that has not expired yet, the transaction is honored and all of the money in the account is withdrawn.

CSRF exploits often involve sites that rely on trust in a user’s identity, often maintained through the use of a cookie. The user’s browser is then tricked into sending HTTP requests to the target site in hopes that a trust between the user’s browser and the target site still exists.

Using CRSF tokens

To stop Cross-site request forgeries from occurring, common practice is to set up the server to generate requests that include a randomly generated parameter with a common name such as "CSRFToken". The token may be generated once per session or a new one generated for each request. If you have used CSRF tokens in your code and enabled CSRF in OpenText DAST, we will take this into consideration when crawling your site. Each time OpenText DAST launches an attack, it will request the form again to acquire a new CSRF token. This adds significantly to the time it take for OpenText DAST to complete a scan, so do not enable CSRF if you are not using CSRF tokens on your site.

Enabling CSRF awareness in OpenText DAST

If your site uses CSRF tokens, you can enable CSRF awareness in OpenText DAST as follows:

1. Select **Default Scan Settings** from the Edit menu.
   The Scan Settings window appears.
2. From the Scan Settings column, select **HTTP Parsing**.
3. Select the **Enable CSRF** box.

Scan settings: Custom Parameters

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **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).

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 OpenText DAST 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 a proprietary OpenText DAST syntax.

Examples:

HTML: User 1 details
Rule: /someDetails/{username}/

HTML: User 1 details
Rule: /TwoParameters/Details/{username}/{parameter2}

HTML: User 1 details
Rule: /{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 web services based on SOAP and Web Services Description Language (WSDL).

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 OpenText DAST how to create the appropriate requests.

Working with rules

You can work with rules for custom parameters as described in the following table.

If you want to...	Then...
Create a rule	1. Click **New Rule**. 2. In the Expression column, enter a rule. See "Path matrix parameters" on the next page for guidelines and examples. The **Enabled** check box is selected by default. OpenText DAST examines the rule and, if it is valid, removes the red X.
Delete a rule	1. Select a rule from the **Custom Parameters Rules** list. 2. Click **Delete**.
Disable a rule without deleting it	1. Select a rule. 2. Clear the check mark in the **Enabled** column.
Import a file containing rules	1. Click  . 2. Using a standard file-selection dialog box, 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 website. If a rule is not invoked during a scan (because the rule doesn't match any URL), then OpenText DAST can programmatically assume that a valid portion of the site has not been attacked. Therefore, if you select this option, OpenText DAST 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, OpenText DAST 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.

Path matrix 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 OpenText DAST Agent
- 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 use special characters to designate parts of the actual URL that contain parameters. If a URL matches a rule, OpenText DAST parses the parameters and attacks them. Notable components of a rule are:

- 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 special elements described in the following table.

Element	Description
*	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 the delimiting brackets that designate 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: /gp/c/*={param}
Invalid rule: /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:

/gp/c/{book_name} is a match for these URLs:

- http://www.amazon.com:8080/gp/c/Moby_Dick
- http://www.amazon.com/gp/c/Singularity_Sky?format=pdf&price=0

- https://www.amazon.com/gp/c/Hobbit

But it is not a match for any of these:

- http://www.amazon.com /gp/c/Moby_Dick/ (no match because of trailing slash)
- http://www.amazon.com/gp/c/Sex_and_the_City/Horror (no match because it has a different number of segments)

OpenText DAST will treat 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, OpenText DAST 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:

- Path – cannot be matched as a whole, since `*` in path matches a single segment or less.
  - Path segments – as in `/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 `/gp/ref=*`, which will match URLs with path containing two segments (first is exactly "gp", second containing any string with prefix "ref=").
- Query – as in `/gp/c/{param}?*`, which matches any URL with 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 `/gp/c/{param}` and `/gp/c/{param}?*`. The first rule will match URL http://www.amazon.com/gp/c/Three_Little_Blind_Mice, while the second will not.
  - Key-value pair of query – as in `/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 `/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 `/gp/c/{param}#*` which matches any URL with fragment part being present

Benefit of using placeholders

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 OpenText DAST and attacked properly.

Multiple rules matching a URL

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}`

OpenText DAST 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.

Scan settings: Filters

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Filters**.

Use the Filters 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 OpenText DAST or those who have access to the raw data or generated reports.

Options

The Filter options are described in the following table.

Option	Description
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.

Adding rules for finding and replacing keywords

Follow the steps below to add a regular expression 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 that describes 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 the section of the request or response you want to search for the filter pattern. The options are:
   - **All** – Search the entire request or response.
   - **Headers** – Search each header individually. Some headers, such as Set-Cookie and HTTP Version headers, are not searched.

     Note: To ensure that all headers are searched, select Prefix.
   - **Post Data** – For requests only, search all of the HTTP message body data.
   - **Body** – Search all of the HTTP message body data.
   - **Prefix** – Simultaneously search everything that is in the request or status line, all headers, and the empty line prior to the body.
4. Type (or paste) the replacement string in the **Replace search text with** box.
   Click ▸ for assistance with regular expressions.
5. For case-sensitive searches, select the **Case sensitive match** check box.
6. Click **OK**.

Scan settings: Cookies/Headers

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Cookies/Headers**.

Standard header parameters

The options in this section are described in the following table.

Option	Description
Include 'referer' in HTTP request headers	Select this check box to include referer headers in OpenText DAST 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 OpenText DAST 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 OpenText DAST 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 OpenText DAST 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.
Accept-Encoding: gzip, deflate	The client requests that the server uses one of the specified encoding methods.

Adding a custom header

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 OpenText DAST to the server when conducting a vulnerability scan.

The default custom cookie used to flag the scan traffic is:

```
CustomCookie=WebInspect;path=/
```

Tip: The equal sign (=) is the delimiter between the name `CustomCookie` and the value `WebInspect`. The `path=/` specifies that the cookie applies to all requests. The custom cookie named `WebInspect` has special processing. Other custom cookies with different names are treated as standard cookies.

Adding a custom cookie

To add a custom cookie:

1. Click **Add**.
   The Specify Custom Cookie window opens.
2. In the **Custom Cookie** box, enter the cookie using the format *<name>=<value>*.

   For example, if you enter

   `CustomCookie=ScanEngine`

   then each HTTP-Request will contain the following header:

   `Cookie: CustomCookie=ScanEngine`

   Tip: If you create a custom cookie and specify the `path=/xyz`, then the custom cookie would only appear in requests starting with "`/xyz`".
3. Click **OK**.

Scan settings: Proxy

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Proxy**.

Options

The Proxy options are described in the following table.

Option	Description
Direct Connection (proxy disabled)	Select this option if you are not using a proxy server.
Auto detect proxy settings	Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.
Use System proxy settings	Import your proxy server information from the local machine.
Use Firefox proxy settings	Import your proxy server information from Firefox. **Note:** Electing to use Firefox proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.
Configure proxy using a PAC file URL	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: 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 **Type** for handling TCP traffic through a proxy server: SOCKS4, SOCKS5, or standard. 3. If authentication is required, select a type from the **Authentication** list: • **Automatic**

Option	Description
	Note: Automatic detection slows the scanning process. If you know and specify one of the other authentication methods, scanning performance is noticeably improved. • **Digest** • **HTTP Basic** • **Kerberos** • **Negotiate** • **NT LAN Manager (NTLM)** 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.

Scan settings: Authentication

To access this feature in a Basic Scan, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **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 is entered whenever OpenText DAST 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 website or application.

Authentication method

If authentication is required, select the authentication method as follows:

- **ADFS CBT**
- **Automatic**
- **Basic**
- **Digest**
- **Kerberos**
- **NT LAN Manager (NTLM)**
- **OAuth 2.0 Bearer**

Authentication credentials

For all authentication methods except OAuth 2.0 Bearer, in the **Authentication Credentials** area:

1. Type a user ID in the **User name** box.
2. Type the user's password in the **Password** box.
3. To guard against mistyping, repeat the password in the **Confirm Password** box.

Caution! OpenText DAST will crawl all servers granted access by this password (if the sites/servers are included in the “allowed hosts” setting). 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 Customer Support.

For the OAuth 2.0 Bearer method, in the **Authentication Credentials** area:

1. Click **Configure**.
   The Open Authorization Configuration dialog opens.
2. Continue with "Configuring OAuth 2.0 bearer credentials" on page 400.

Client certificates

Client certificate authentication allows users to present client certificates rather than entering a user name and password. You can select a certificate from the local machine or a certificate assigned to a current user. You can also select a certificate from a mobile device, such as a common access card (CAC) reader that is connected to your computer. To use client certificates:

1. In the Client Certificates area, select the **Enable** check box.
2. Click **Select**.

The Client Certificates window opens.

3. Do one of the following:
   - To use a certificate that is local to the computer and is global to all users on the computer, select **Local Machine**.
   - To use a certificate that is local to a user account on the computer, select **Current User**.

     Note: Certificates used by a common access card (CAC) reader are user certificates and are stored under Current User.

4. Do one of the following:
   - To select a certificate from the "Personal" ("My") certificate store, select **My** from the drop-down list.
   - To select a trusted root certificate, select **Root** from the drop-down list.
5. Does the website use a CAC reader or a certificate that is password protected?
   - If *yes*, do the following:
     i. Select a certificate that is prefixed with “(Protected)” from the **Certificate** list.

     Information about the selected certificate and a Password/PIN field appear in the Certificate Information area.

     ii. If a password or PIN is required, type it in the **Password/PIN** field.

     Note: If a password or PIN is required and you do not enter it at this point, you must enter the password or PIN in the Windows Security window each time it prompts you during the scan.

     Important! By default, OpenText DAST uses OpenSSL. If you are using a specific SSL/TLS protocol rather than OpenSSL, the Profiler portion of scan configuration may not work with certificates that are protected with a password.

     iii. Click **Test**.

     If you entered the correct password or PIN, a Success message appears.
   - If *no*, select a certificate from the **Certificate** list.

     Information about the selected certificate appears below the Certificate list.
6. Click **OK**.

Updating certificates in composite scan settings

The composite scan settings include a BIN file that contains encrypted certificate data. If you need to replace or update the client certificate in your composite scan settings, you can place the updated PFX or P12 file inside the certificates directory in the composite settings ZIP file. When OpenText DAST opens the settings, it will check for PFX and P12

files first. If none are present, then the BIN file will be decrypted and used. For more information about composite settings, see "Application settings: General" on page 431.

To replace or update the client certificate:

1. Locate the encrypted BIN file in the certificates directory in your composite scan settings ZIP. The file name is a GUID, similar to the following:

   `<your-scansettings.zip>\certificates\0b627638-efda-4d01-a83e-80ee3a79b4cf.bin`

   Note: The default settings file location in Windows is `C:\ProgramData\HP\HP WebInspect\Settings\`.

2. Place the updated PFX or P12 file into the same directory.
3. Rename the PFX or P12 file the same name as the BIN file. Using the previous example, the file name will be as follows:

   `0b627638-efda-4d01-a83e-80ee3a79b4cf.pfx`

   – OR –

   `0b627638-efda-4d01-a83e-80ee3a79b4cf.p12`

   Important! Be sure to retain the original file extension.

4. Optionally, if you want to retain an encrypted certificate in the settings, save the settings again and the BIN file will reflect the updated PFX or P12 certificate. The PFX or P12 certificates will be removed from the ZIP.

Tip: PFX and P12 certificates frequently require a password. Use one of the following options to provide the password for the settings:

- Set an empty password when you create the PFX or P12 certificate and place it in the settings ZIP file.
- Keep the password for the PFX or P12 certificate and edit the `settings.json` file to place the password as the value for the CertificatePin as follows:

  `"CertificatePin": "<password>"`

Editing the proxy config file for OpenText DAST tools

When using tools that incorporate a proxy (specifically Web Macro Recorder, Web Proxy, 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 perform the following tasks to edit the `SPI.Net.Proxy.Config` file.

Task 1: Find your certificate's serial number

1. Open the Microsoft Edge browser.
2. Search the settings for "Internet Explorer compatibility" and enable the setting, if necessary.
3. Select **Internet Options**.

4. On the Internet Options window, select the **Content** tab and click **Certificates**.
5. On the Certificates window, select a certificate and click **View**.
6. On the Certificate window, click the **Details** tab.
7. Click the **Serial Number** field and copy the serial number that appears in the lower pane (highlight the number and press Ctrl + C).
8. 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\Fortify\Fortify 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\.spidynamics\.com`).

   Number is the serial number obtained in Task 1.
3. Save the edited file.

Enable macro validation

Most dynamic application scans require user authentication to expose the complete surface of the application. Failure of the login macro to log in to the application results in a poor quality scan. If the login macro quality is measured before the scan, then low quality scans can be avoided.

Select **Enable macro validation** to enable OpenText DAST to test for inconsistencies in macro behavior at the start of the scan. For more information about the specific tests performed, see "Testing login macros" on page 494.

Use a login macro for forms authentication

This type of macro is used primarily for Web form authentication. It incorporates logic that will prevent OpenText DAST 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 ellipsis button  to locate the macro. Click **Record** to record a macro.

Note: The Record button is not available for Guided Scan, because Guided Scan includes a separate stage for recording a login macro.

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 username and password parameters.

If you start a scan using a macro that includes parameters for user name and password, then when you scan the page containing the input elements associated with these entries, OpenText DAST substitutes the user name and password specified here. This feature enables 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. This also applies to parameters for phone number, email, and email password that are used in two-factor authentication scans.

If the macro uses parameters for which values are masked in the Web Macro Recorder, then these values are also masked when configuring a Basic Scan or Guided Scan in OpenText DAST.

For information about creating parameters using the Web Macro Recorder, see the Web Macro Recorder chapters in the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Use a startup macro

This type of macro is used most often to focus on a particular subsection of the application. It specifies URLs that OpenText DAST will use to navigate to that area. It may also include login information, but does not contain logic that will prevent OpenText DAST from logging out of your application. OpenText DAST visits all URLs in the macro, collecting hyperlinks and mapping the data hierarchy. It then calls the Start URL and begins a normal crawl (and, optionally, audit). Click the ellipsis button to locate the macro. Click **Record** to record a macro.

Multi-user login

You can use the Multi-user Login option to parameterize the username and password in a login macro, and then define multiple username and password pairs to use in a scan. You can also parameterize the phone number, email, and email password if two-factor authentication is required. This approach allows the scan to run across multiple threads. Each thread has a different login session, resulting in faster scan times.

Important! To use Multi-user Login, you must first select **Use a login macro for forms authentication** and record a new macro or select an existing macro to use. See "Use a login macro for forms authentication" on the previous page.

To use multiple user logins to conduct the scan:

1. Select the **Multi-user Login** checkbox.

   Note: If you clear the Multi-user Login checkbox prior to running the scan, the additional credentials will not be used during the scan. OpenText DAST will use only the original credentials recorded in the login macro.

2. Continue according to the following table:

To...	Then...
Add a user's credentials	a. Under Multi-user Login, click **Add**. The Multi-user Credential Input dialog box appears. b. In the **Username** field, type a username c. In the **Password** field, type the corresponding password. d. Optionally, if two-factor authentication is required, then add the following criteria: ◦ **Phone Number** - corresponding phone number for the username (to receive SMS responses) ◦ **Email** - corresponding email address for the username (to receive email responses) ◦ **Email Password** - password for the email address (to receive email responses) e. Click **OK**. f. Repeat Steps a-e for each user login to add. **Important!** The number of shared requestor threads should not be more than the number of configured users. Requestor threads without valid users will cause the scan to run longer. Remember to count the original username and password in the parameterized macro as the first user when you configure multiple users. For more information, see "Scan settings: Requestor" on page 369.
Edit a user's credentials	a. Under Multi-user Login, select an entry in the table and click **Edit**. The Multi-user Credential Input dialog box appears. b. Edit the credentials as needed. c. Click **OK**.
Delete a user's	a. Under Multi-user Login, select an entry in the table to be

To...	Then...
credentials	removed. b. Click **Delete**.

For more information, see "Multi-user login scans" on page 189.

Configuring OAuth 2.0 bearer credentials

Open authorization (OAuth) 2.0 is an open-standard authorization protocol that shares authorization tokens between services or applications to prove the identity of a user. You can configure the following types of OAuth 2.0 authentication flows in the Open Authorization Configuration dialog:

- **Client Credentials Grant** – The client uses its client credentials, such as client ID and client secret, when requesting access to the protected resources.
- **Password Credentials Grant** – The client obtains the resource owner's credentials, such as username and password, usually by way of an interactive form.

If you configure OAuth 2.0 authentication, then OpenText DAST will use the retrieved token for the entire scan. The token will be refreshed if it expires.

To configure OAuth 2.0 authentication in the Open Authorization Configuration dialog:

1. In the **OAuth flows** list, select a flow. Options are **Client Credentials Grant** and **Password Credentials Grant**.
2. In the **Access Token URL** box, type the URL that is used to generate tokens, such as `https://<yourDomain>/oauth2/token`.
3. Optionally, if your service supports different scopes (or permissions) for the OAuth flow, double-click the value box for the **scope** parameter and specify the scope to use.

   Tip: If a parameter is unneeded, you can leave the value empty or select the parameter row and press the **delete** key to remove it.

4. Provide information that will be included in the authorization request header according to the following table.

To configure...	Then...
A Client Credentials Grant flow	a. Double-click the value box for the **client_id** parameter and enter the application (client) ID. b. Double-click the value box for the **client_**

To configure...	Then...
	secret parameter and enter the client secret that you generated for your application in the OAuth provider's registration portal.
A Password Credentials Grant flow	a. Double-click the value box for the **username** parameter and enter the username. b. Double-click the value box for the **password** parameter and enter the password.

Tip: Optionally, you can double-click an empty row and add custom parameter names and values. However, be aware of the following restrictions:

- The `grant_type` and `scope` parameter names are reserved and cannot be used in a custom parameter.
- If the OAuth Flow Type is Client Credentials Grant, then `client_credentials`, `client_id`, and `client_secret` cannot be used in a custom parameter.
- If the OAuth Flow Type is Password Credentials Grant, then `username` and `password` cannot be used in a custom parameter.

5. By default, OpenText DAST uses Status Code 403 for the logout signature. Optionally, if you use a custom status code, in the **Logout Signature** box, enter the status code or a regular expression to indicate the logout signature. Use the following syntax:

   `[STATUSCODE]<Number>`
6. Optionally, click **Test** to test access to the server and receipt of a bearer token.

   Note: To view the bearer token or to see any errors, click **Show Details** in the Test Results dialog. The **Expires In** value shown in the details indicates how long the token is valid. The token will be refreshed after it has expired.
7. Click **OK**.

Scan settings: File Not Found

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **File Not Found**.

Options

The File Not Found options are described in the following table.

Option	Description
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. You can then identify the codes that fit the following categories: • **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. OpenText DAST 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 FNF 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 OpenText DAST from 404 pages that are unique to your site.
Auto detect FNF page	Some websites 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 OpenText DAST to detect these "custom" file-not-found pages. OpenText DAST 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 FNF page** check box, you can specify what percentage of the response content must be the same in the **Match FNF page with** field. The default is 90 percent.

Scan settings: Policy

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **Policy**.

You can change to 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 select multiple policies which will be aggregated by the sensor during the scan. For descriptions of policies, see "OpenText DAST policies" on page 458.

You can also create, import, or delete custom policies.

Selecting one or more policies

To select a different policy:

1. In the **Audit Policies** list, slide the toggle for the selected policy to the disabled position.
2. Slide the toggle for the desired policy to the enabled position.
3. Click **OK**.

To select additional policies:

1. In the **Audit Policies** list, slide the toggle for the desired policies to the enabled position.
2. Click **OK**.

Creating a policy

To create a policy:

1. Click **Create**.
   The Policy Manager tool opens.
2. Select **New** from the **File** menu (or click the New Policy icon).
3. Select the policy on which you will model a new one.
4. Refer to the Policy Manager documentation for additional instructions.

Editing a policy

To edit a policy:

1. Select a custom policy.

   Note: Only custom policies may be edited.

2. Click **Edit Scan Policy**.
   The Policy Manager tool opens.
3. Refer to the Policy Manager documentation for additional instructions.

Importing a policy

To import a policy:

1. Click **Import**.
2. On the Import Custom Policy window, click the ellipses 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 OpenText DAST.
   - All Files (*.*): Files of any type, including non-policy files.
4. Click **OK**.

A copy of the policy is created in the Policies folder (the default location is `C:\ProgramData\HP\HP WebInspect\Policies\`). The policy and all of its enabled checks are imported into SecureBase using the specified policy name. Custom agents are not imported.

Deleting a policy

To delete a policy:

1. Select a custom policy.

   Note: Only custom policies may be deleted.

2. Click **Delete Scan Policy**.
   A confirmation message appears.
3. Click **Yes**.
   The policy is removed from your Policies directory.

Scan settings: User Agent

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Scan Settings** category, select **User Agent**.

You can configure user agent settings that synchronize in both OpenText DAST and the Event-based Web Macro Recorder.

Profile and user-agent string

You can select a predefined Profile that specifies the user agent string for the browser. The following table describes the available profiles.

Profile	User-Agent String
Default	`Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0`
Google Chrome (Windows)	`Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36`
Microsoft Edge (Windows)	`Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36 Edg/133.0.0.0`
Safari (macOS)	`Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3 Safari/605.1.15`
Googlebot 2.1	`Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)`
Bingbot	`Mozilla/5.0 (compatible; bingbot/2.0; +http://www.bing.com/bingbot.htm)`
Yahoo! Slurp	`Mozilla/5.0 (compatible; Yahoo! Slurp; http://help.yahoo.com/help/us/ysearch/slurp)`
Safari (iOS)	`Mozilla/5.0 (iPhone; CPU iPhone OS 18_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3 Mobile/15E148 Safari/604.1`
Safari (iPadOS)	`Mozilla/5.0 (iPad; CPU OS 18_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3 Mobile/15E148 Safari/604.1`
Google Chrome (Android)	`Mozilla/5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Mobile Safari/537.36`
Custom	User-specified. **Note:** User agents from earlier versions of OpenText DAST that

Profile	User-Agent String
	have been removed, such as Internet Explorer 6, Internet Explorer 7, Internet Explorer 8, and iPhone, iOS 14.3, are considered a Custom profile in existing scans. **Important!** OpenText recommends that only advanced users specify a Custom profile.

Navigator interface settings

The Navigator Interface settings provide information that legacy web applications use to facilitate browser detection. You can customize these settings if you require browser-specific behavior. The following table describes these settings for each user agent profile.

Profile	appName	appVersion	platform[1]
Default	`Netscape`	`5.0 (Windows)`	`Win64`
Google Chrome (Windows)	`Netscape`	`5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36`	`Win32`
Microsoft Edge (Windows)	`Netscape`	`5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36 Edg/133.0.0.0`	`Win32`
Safari (macOS)	`Netscape`	`5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3 Safari/605.1.15`	`MacIntel`
Googlebot 2.1	N/A	N/A	N/A
Bingbot	N/A	N/A	N/A
Yahoo! Slurp	N/A	N/A	N/A
Safari (iOS)	`Netscape`	`5.0 (iPhone; CPU iPhone OS 18_3 like Mac OS X) AppleWebKit/605.1.15`	`iPhone`

[1]The browser returns an empty string or a string representing the platform on which the browser is running.

Profile	appName	appVersion	platform[1]
		(KHTML, like Gecko) Version/18.3 Mobile/15E148 Safari/604.1	
Safari (iPadOS)	Netscape	5.0 (iPad; CPU OS 18_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/18.3 Mobile/15E148 Safari/604.1	iPad
Google Chrome (Android)	Netscape	5.0 (Linux; Android 10; K) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Mobile Safari/537.36	Linux armv81

[1]The browser returns an empty string or a string representing the platform on which the browser is running.

Chapter 7: Crawl settings

This chapter describes the crawl settings that are used by the OpenText DAST crawler. The OpenText DAST crawler is a software program designed to follow hyperlinks throughout a website, retrieving and indexing pages to document the hierarchical structure of the site. The parameters that control the manner in which OpenText DAST crawls a site are available from the Crawl Settings list.

Crawl settings: DOM Coverage

OpenText DAST can use your site's Document Object Model (DOM) when crawling the site.

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Crawl Settings** category, select **DOM Coverage**.

Setting a DOM Coverage limit

1. Select **Single limit**.
   This is the default option.
2. In the **Limiters** area, enter a value in the **Maximum script events per page**. The default value is `1000`.
3. Click **OK**.

Setting a DOM Coverage strategy

To turn on DOM Coverage and add rules:

1. Select **Manual configuration**.
   The **Manual rules** area appears.
2. In the **Limiters** area, enter a value in the **Maximum script events per page**. The default value is `1000`.
3. In the **Manual Rules** area, click **Add**.
   The **Edit Rule** dialog box opens.
4. In the **Edit Rule** dialog box:
   a. Enter a **Rule Name**.
   b. In the **Limiters** area, click **Add**.
      The **Add Limiter** dialog box opens. Enter the **XPath** of the page element and script events **Limit**. Click **OK** when done.

In the **Conditions** area, click **Add**.

The **Add Condition** dialog box opens. Enter the **XPath** of the page element. Click **OK** when done.

c. Click **OK** to close the **Edit Rule** dialog box.

5. Click **OK** to close the **Default Settings** window.

Crawl settings: Link Parsing

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Crawl Settings** category, select **Link Parsing**.

OpenText DAST follows all hyperlinks defined by HTML (using the `<a href>` tag) and those defined by scripts (JavaScript and VBScript). However, you may encounter other communications protocols that use a different syntax for specifying links. To accommodate this possibility, you can use the Custom Links feature and regular expressions to identify links that you want OpenText DAST to follow. These are called special link identifiers.

Adding a specialized link identifier

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**.

Crawl settings: Link Sources

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Crawl Settings** category, select **Link Sources**.

What is link parsing?

The OpenText DAST crawler sends a request to a start URL and recursively parses links (URLs) from the response content. These links are added to a work queue and the crawler iterates through the queue until it is empty. The techniques used to extract the link information from the HTTP responses are collectively referred to as 'link parsing.' There are two choices for how the crawler performs link parsing: Pattern-based and DOM-based.

Pattern-based parsing

Pattern-based link parsing uses a combination of text searching and pattern matching to find URLs. These URLs include the ordinary content that is rendered by a browser, such as <A> elements, as well as invisible text that may reveal additional site structure.

This option matches the default behavior of OpenText DAST version 10.40 and earlier. This is a more aggressive approach to crawling the website and can increase the amount of time it takes to conduct a scan. The aggressive behavior can cause the crawler to create many extra links which are not representative of actual site content. For these situations, DOM-based parsing should expose the site's URL content with fewer false positives.

Note: All of the DOM-based Parsing techniques for finding links are used when Pattern-based Parsing is selected. Pattern-based Parsing, however, is not capable of computing the metadata for the link source. DOM-based Parsing is capable of computing this information and thus provides more intelligent parsing. DOM-based Parsing also provides more control over which parsing techniques are used.

DOM-based parsing

The Document Object Model (DOM) is a programming concept that provides a logical structure for defining and building HTML and XML documents, navigating their structure, and editing their elements and content.

A graphical representation of an HTML page rendered as DOM would resemble an upside-down tree: starting with the HTML node, then branching out in a tree structure to include the tags, sub-tags, and content. This structure is called a DOM tree.

Using DOM-based parsing, OpenText DAST parses HTML pages into a DOM tree and uses the detailed parsed structure to identify the sources of hyperlinks with higher fidelity and greater confidence. DOM-based parsing can reduce false positives and may also reduce the degree of 'aggressive link discovery.'

On some sites, the crawler iteratively requests bad links and the resulting responses echo those links back in the response content, sometimes adding extra text that compounds the problem. These repeated cycles of 'bad links in and bad links out' can cause scans to run for a long time or, in rare cases, forever. DOM-based parsing and careful selection of link sources provide a mechanism for limiting this runaway scan behavior. Web applications vary in structure and content, and some experimentation may be required to get optimal link source configurations.

To refine DOM-based Parsing, select the techniques you want to use for finding links. Clearing techniques that may not be a concern for your site may decrease the amount of time it takes to complete the scan. For a more thorough scan, however, select all techniques or use Pattern-based Parsing. The DOM-based Parsing techniques are described in the following table. For more information, see "Limitations of link source settings" on page 416.

Technique	Description
Include Comment Links (Aggressive)	Programmers may leave notes to themselves that include links inside HTML comments that are not visible on the site, but may be discovered by an attacker. Use this option to find links inside HTML comments. OpenText DAST will find more links, but these may not always be valid URLs, causing the crawler to try to access content that does not exist. Also, the same link can be on every page and those links can be relative, which can exponentially increase the URL count and lengthen the scan time.
Include Conditional Comment Links	A conditional comment link occurs when the HTML on the page is conditionally included or excluded depending on the user agent (browser type and version) making the request. **Regular comment example:** `<!—hidden.txt -->` **Conditional comment example:** `<!--[if lt IE9]>` `<script src="//www.somesite.com/static/v/all/js/html5sh.js"></script>` `<link rel="stylesheet" type"text/css" href='//www.somesite.com/static/v/fn-hp/css/IE8.css'>` `<![endif]-->` OpenText DAST emulates browser behaviors in evaluating HTML code and processes the DOM differently depending on the user agent. A link found in a comment by one user agent is a normal HTML link for other user agents. Use this option to find conditional links that are inside HTML commands, such as those commented out based on browser version. These conditional statements may also contain script includes that need to be executed when script parsing is enabled. Crawling these links will be more thorough, but can increase the scan time. Additionally, such comments may be out of date and pointless to crawl.
Include Plain Text	Plain text in a .txt file or a paragraph inside HTML code can be formatted

Technique	Description
Links	as a URL, such as `http://www.something.com/mypage.html`. However, because this is only text and not a true link, the browser would not render it as a link, and the text would not be functionally part of the page. For example, the content may be part of a page that describes how to code in HTML using fake syntax that is not meant to be clicked by users. Use this option for OpenText DAST to parse these text links and queue them for a crawl. Also, using smart pattern matches, OpenText DAST can identify common file extensions, such as .css, .js, .bmp, .png, .jpg, .html, etc., and add these files to the crawl queue. Auditing these files that are referenced in plain text can produce false positives.
Include Links in Static Script blocks	Use this option for OpenText DAST to examine inside the opening and closing script tags for text that looks like links. Valid links may be found inside these script blocks, but developers may also leave comments that include text resembling links inside the opening and closing script tags. For example: `<script type="text/javascript">` `// go to http://www.foo.com/blah.html for help` `var url = "http:www.foo.com/xyz/" + path + "?help"` `</script>` Additionally, JavaScript code inside these tags can be handled by the JavaScript execution engine during the scan. However, searching for static links in a line of code that sets a variable, such as the “var url” in the example above, can create problems when those partial paths are added to the queue for crawling. If the variable includes a relative link with a common extension, such as “foo.html”, the crawler will append the extension to the end of every page that includes the line of code. This can produce unusable URLs and may create false positives.
Parse URLs Embedded in URLs	Use this option for OpenText DAST to parse any text that is inside an href attribute and add it to the crawl queue. The following is an example of a URL embedded in a URL: `<a` `href="http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz` `.com%2Fblah" />` On some sites, however, file not found pages return the URL in a form action tag and append the URL to the original URL as follows: `<form`

Technique	Description
	`action="http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz.com%2Fblah? http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz.com%2Fblah" />` OpenText DAST will then request the form action, and receive another file not found response, again with the URL appended in a form action, as shown below: `<form action="http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz.com%2Fblah? http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz.com%2Fblah? http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz.com%2Fblah? http://www.foo.com/xyz/bar.html?url=http%3A%2F%2Fwww.zzzz.com%2Fblah" />` On such a site, these URLs will continue to produce file not found responses that add more URLs to the crawl queue, creating an infinite crawl loop. To avoid adding this type of URL to the crawl queue, do not use this option.
Allow Un-rooted URLs (for the above items)	This option modifies the behavior of the previous five options. Some URLs do not include the specific scheme, such as http, and are not fully qualified domain names. These URLs, which may resemble `xyz.html`, are considered unanchored or “un-rooted.” The assumption is that the un-rooted URL is relative to the request. For example, the non-fully qualified URL `<a href='foo.html' />` does not include a scheme. This URL uses the scheme of the context URL. If an HTTPS page requested to get the content, then HTTPS would be prepended to the URL. Use this option to treat un-rooted URLs as links when parsing. If this option is selected, the scan will be more thorough and more aggressive, but may take considerably longer to complete. **URL Samples and Parsing Results** The following samples describe various URLs and how they are parsed during a crawl. **A Normal URL**

Technique	Description
	The URL in the following request includes a forward (or anchor) slash. Request from `http://www.foo.com/x/y/z/` For `<a href='/bar.html' />` Results in a link to `http://www.foo.com/bar.html`. **Simple Un-rooted URL** The URL in the following request is un-rooted because it does not include a forward slash. Request from `http://www.foo.com/` For `<a href='bar.html' />` Results in a link to `http://www.foo.com/bar.html`. **Long Un-rooted URL** The following request shows a long, un-rooted URL. Request from `http://www.foo.com/x/y/z/` For `<a href='bar.html' />` Results in a link to `http://www.foo.com/x/y/z/bar.html`. **Comments in Code** You may include comments, such as `<!-- baz_ads.js -->`, in your code before a script include. The following request shows how this comment is interpreted during an aggressive crawl. Request from `http://www.foo.com/x/y/z/` For `<!-- baz_ads.js -->` Results in a link to `http://www.foo.com/x/y/z/baz_ads.js` If you include this comment on your master page, then during an aggressive scan, the comment will be discovered on many, if not all, page responses in the site. This configuration can cause runaway scans. The comment `<!-- baz_ads.js -->` on the master page results in multiple links: `http://www.foo.com/baz_ads.js` `http://www.foo.com/x/baz_ads.js` `http://www.foo.com/x/y/baz_ads.js` `http://www.foo.com/x/y/z/baz_ads.js` And so on for all pages in the site.

Form actions, script includes, and stylesheets

Some link types–such as form actions, script includes, and stylesheets–are special and are treated differently than other links. On some sites, it may not be necessary to crawl and parse these links. However, if you want an aggressive scan that attempts to crawl and parse everything, the following options will help accomplish this goal. For more information, see "Limitations of link source settings" on the next page.

Note: You can also allow un-rooted URLs for each of these options. See “Allow Un-rooted URLs” in this topic.

Option	Description
Crawl Form Action Links	When OpenText DAST encounters HTML forms during the crawl, it creates variations on the inputs that a user can make and submits the forms as requests to solicit more site content. For example, for forms with a POST method, OpenText DAST can use a GET instead and possibly reveal information. In addition to this type of crawling, use this option for OpenText DAST to treat form targets as normal links.
Crawl Script Include Links	A script include imports JavaScript from a .js file and processes it on the current page. Use this option for OpenText DAST to crawl the .js file as a link.
Crawl Stylesheet Links	A stylesheet link imports the style definitions from a .css file and renders them on the current page. Use this option for OpenText DAST to crawl the .css file as a link.

Miscellaneous options

The following additional options may help improve link parsing for your site. For more information, see "Limitations of link source settings" on the next page.

Option	Description
Crawl Links on FNF Pages	If you select this option, OpenText DAST will look for and crawl links on responses that are marked as “file not found.” This option is selected by default when the Scan Mode is set to Crawl Only or Crawl & Audit. The option is not available when the Scan Mode is set to Audit Only.

Option	Description
Suppress URLs with Repeated Path Segments	Many sites have text that resembles relative paths that become unusable URLs after OpenText DAST parses them and appends them to the URL being crawled. These occurrences can result in a runaway scan if paths are continuously appended, such as `/foo/bar/foo/bar/`. This setting helps reduce such occurrences and is enabled by default. With the setting **enabled**, the options are: **1** – Detect a single sub-folder repeated anywhere in the URL and reject the URL if there is a match. For example, `/foo/baz/bar/foo/` will match because "`/foo/`" is repeated. The repeat does not have to occur adjacently. **2** – Detect two (or more) pairs of adjacent sub-folders and reject the URL if there is a match. For example, `/foo/bar/baz/foo/bar/` will match because "`/foo/bar/`" is repeated. **3** – Detect two (or more) sets of three adjacent sub-folders and reject the URL if there is a match. **4** – Detect two (or more) sets of four adjacent sub-folders and reject the URL if there is a match. **5** – Detect two (or more) sets of five adjacent sub-folders and reject the URL if there is a match. If the setting is **disabled**, repeating sub-folders are not detected and no URLs are rejected due to matches.

Limitations of link source settings

Clearing a link source check box prevents the crawler from processing that specific kind of link when it is found using static parsing. However, these links can be found in many other ways. For example, clearing the **Crawl Stylesheet Links** option does not control path truncation nor suppress .css file requests made by the script engine. Clearing this setting only prevents static link parsing of the .css response from the server. Similarly, clearing the **Crawl Script Include Links** option does not suppress .js, AJAX, frameIncludes, or any other file request made by the script engine. Therefore, clearing a link source check box is not a universal filter for that type of link source.

The goal for clearing a check box is to prevent potentially large volumes of bad links from cluttering the crawl and resulting in extremely long scan times.

Crawl settings: 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**) enables 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.

Adding a file extension to exclude/reject

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.

Adding a MIME type to exclude

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. 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** - OpenText DAST 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, OpenText DAST will not examine the specified URL or host for links to other resources. During the audit portion of the scan, OpenText DAST 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.

Editing the default criteria

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**.

Adding exclusion/rejection criteria

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 Fortify'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  (or press **Enter**).
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 **Other Exclusion/Rejection Criteria** list, select either **Reject**, **Exclude**, or both.

Note: You cannot reject Response, Response Header, and Status Code Target types during a scan. You can only exclude these Target types.

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, OpenText DAST 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]*/

Chapter 8: Audit settings

This chapter describes the audit settings used by OpenText DAST during an audit scan. An audit is the probe or attack conducted by OpenText DAST which is designed to detect vulnerabilities. The parameters that control the manner in which OpenText DAST conducts that probe are available from the Audit Settings list.

Audit settings: Session Exclusions

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Audit Settings** category, select **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**) enables you to specify additional objects to be excluded from the audit.

Excluded or rejected file extensions

If you select **Reject**, OpenText DAST will not request files having the specified extension.

If you select **Exclude**, OpenText DAST will request files having the specified extension, but will not audit them.

Adding a file extension to exclude/reject

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

OpenText DAST will not audit files associated with the MIME types you specify.

Adding a MIME type to exclude

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. 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** - OpenText DAST 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, OpenText DAST will not examine the specified URL or host for links to other resources. During the audit portion of the scan, OpenText DAST 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.

Editing the default criteria

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**.

Adding exclusion/rejection criteria

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 Fortify'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  (or press Enter).
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 **Other Exclusion/Rejection Criteria** list, select either **Reject**, **Exclude**, or both.

Note: You cannot reject Response, Response Header, and Status Code Target types during a scan. You can only exclude these Target types.

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, OpenText DAST 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]*/

Audit settings: Attack Exclusions

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Audit Settings** category, select **Attack Exclusions**.

Excluded parameters

Use this feature to prevent OpenText DAST 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.

Adding parameters to exclude

To prevent certain parameters from being modified:

1. In the **Excluded Parameters** group, click **Add**.
   The Specify HTTP Exclusions 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 OpenText DAST 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, the name of the cookie is "FirstCookie."

```
Set-Cookie: FirstCookie=Chocolate+Chip; path=/
```

Excluding certain cookies

To exclude certain cookies:

1. In the **Excluded Headers** group, click **Add**.
   The Regular Expression Editor appears.

   Note: 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?
   - If *yes*, click **OK**.
   - If *no*, verify that the Comparison Text contains the string you want to identify or modify the regular expression.

Excluded headers

Use this feature to prevent OpenText DAST from using certain headers in the HTTP request to attack the Web site. This feature is used to avoid corrupting header values.

Excluding certain headers

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.

   Note: 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?

- If *yes*, click **OK**.
- If *no*, verify that the Comparison Text contains the string you want to identify or modify the regular expression.

Audit Inputs Editor

Using the Audit Inputs Editor, you can create or modify 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**.

Audit settings: Attack Expressions

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Audit Settings** category, select **Attack Expressions**.

Additional regular expression languages

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 - China
- zh-tw: Chinese - Taiwan
- ja-jp: Japanese - Japan
- ko-kr: Korean - Korea
- pt-br: Portuguese - Brazil
- es-es: Spanish - Spain

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.

Audit settings: Vulnerability Filtering

To access this feature, click the **Edit** menu and select **Default Settings** or **Current Settings**. Then, in the **Audit Settings** category, select **Vulnerability Filtering**.

By applying certain filters, you can limit the display of certain vulnerabilities reported during a scan. The options are:

- **Standard Vulnerability Definition** - This filter sorts parameter names for determining equivalency between similar requests. For example, if a SQL injection vulnerability is found in parameter "a" in both `http://x.y?a=x;b=y` and `http://x.y?b=y;a=x`, it would be considered equivalent.
- **Parameter Vulnerability Roll-Up** - This filter consolidates multiple parameter manipulation and parameter injection vulnerabilities discovered during a single session into one vulnerability.
- **403 Blocker** - This filter revokes vulnerabilities when the status code of the vulnerable session is 403 (Forbidden).
- **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 vulnerability filter

To add a filter to your default settings:

1. Click the **Edit** menu and select **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** list and click **Remove**.

   The filter is removed from the **Enabled Filters** list and added to the **Disabled Filters** list.

You can also modify the settings for a specific scan by clicking the **Settings** button at the bottom of the Scan Wizard or the Web Service Scan Wizard.

Suppressing off-site vulnerabilities

If your web application includes links to hosts that are not in your Allowed Hosts list, OpenText DAST may identify passive vulnerabilities on those hosts. To suppress all vulnerabilities against sessions for off-site hosts that are not in your Allowed Hosts list, select the **Suppress Offsite Vulnerabilities** check box.

For more information about Allowed Hosts, see "Scan settings: Allowed Hosts" on page 375.

Audit settings: Smart Audit

To access this feature, click the **Edit** menu and select **Default Scan Settings** or **Current Scan Settings**. Then, in the **Audit Settings** category, select **Smart Audit**.

Enable Smart Audit

Smart Audit is an "intelligent" feature that discovers the type of server that is hosting the website and checks for known vulnerabilities against that specific server type. For example, if you are scanning a site hosted on an IIS server, OpenText DAST 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 this option, 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 OpenText DAST, 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.

To specify a custom definition:

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**.

   OpenText DAST 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.

Note: 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**.

Chapter 9: Application settings

This chapter describes the settings that define where OpenText DAST stores scan data and log files, as well as settings for licensing and SmartUpdates. These settings also configure OpenText DAST to interact with other applications, such as OpenText Application Lifecycle Management (ALM).

Application settings: General

To access this feature, click **Edit > Application Settings** and then select **General**.

General

The General options are described in the following table.

Option	Description
Enable Active Content in Browser Views	Select this option to allow execution of JavaScript and other dynamic content in all browser windows within OpenText DAST. For example, one OpenText DAST 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:  **Note:** If you initiate or open a scan while this option is disabled, and you then enable this option, the browser will not execute the active content until you close and then reopen the scan.

Option	Description
Enable Diagnostic File Creation	If the OpenText DAST application should ever fail, this option forces OpenText DAST to create a file containing data that was stored in main memory at the time of failure. You can then provide the file to OpenText support personnel. 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, OpenText DAST displays various prompts and dialog boxes to remind you of certain consequences that may occur as a result of an action you take. These dialog boxes contain a check box labeled "Don't show me again." If you select that option, OpenText DAST discontinues displaying those messages. You can force OpenText DAST to resume displaying those messages if you click **Reset "Don't Show Me Again" messages**.
Use Seven Pernicious Kingdom (7PK) Taxonomy	This option enables you to select The Seven Pernicious Kingdoms taxonomy for ordering and organizing the reported vulnerabilities. Seven Pernicious Kingdoms (7PK) is a taxonomy of software security errors developed by the OpenText 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

Option	Description
	4. Time and State 5. Errors 6. Code Quality 7. Encapsulation * Environment The first seven kingdoms are associated with security defects in source code, while the last one describes security issues outside the actual code. 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 https://vulncat.fortify.com/. You might want to use the Seven Pernicious Kingdoms taxonomy if you are integrating OpenText DAST with other OpenText Fortify products as it provides for a unified taxonomy.
Use OpenSSL Engine	By default, OpenText DAST uses this option. The OpenSSL engine provides support for websites that require use of the TLS 1.3 security protocol. OpenSSL is backwards compatible with previous versions of the TLS protocol. If this option is enabled, the SSL/TLS Protocols options are disabled in Scan settings: Method. You cannot select individual protocols for a scan.
Enable HTTP/2 Support	Use this option if your website supports the HTTP/2 protocol *only* and you experience issues using the HTTP/1 protocol.
Use Composite Scan Settings	Composite settings consist of a JSON version of the scan settings packaged in a ZIP file with any binary files required for the scan, such as macros, client certificates, custom policies, and so forth. OpenText ScanCentral DAST uses composite settings. By default, OpenText DAST uses XML settings. Selecting this option enables OpenText DAST to create and use composite settings that are ready to import and use in OpenText

Option	Description
	ScanCentral DAST without conversion. You cannot convert XML settings into composite settings in the OpenText DAST UI. You must use the OpenText DAST API endpoint `configuration/scansettings/convert` or import the settings into OpenText ScanCentral DAST for automatic conversion. Additionally, when saving a settings file from a scan that was conducted using XML settings, the file is saved in the XML format even if the Use Composite Scan Settings option is enabled. **Important!** If you enable this option, only ZIP settings will be accessible in OpenText DAST for configuration purposes. To access XML settings, you must disable this option.

OpenText DAST Agent

The OpenText DAST Agent options are described in the following table.

Option	Description
Use WebInspect Agent information when encountered on target site	If this option is selected and OpenText DAST detects that OpenText DAST Agent is installed on a target server, it will incorporate OpenText DAST Agent information to improve overall scan efficiency. A notation on the OpenText DAST dashboard indicates whether or not OpenText DAST Agent has been detected.
Automatically group by duplicate vulnerabilities in vulnerability window	If this option is selected and OpenText DAST Agent information is used (above setting), then vulnerabilities listed on the **Findings** tab in the Summary pane will be grouped by check and then by equivalent vulnerabilities.
Allow WebInspect Agent to suggest attack strategy	If this option is selected and OpenText DAST information is used (see *Use WebInspect Agent Information When Encountered on Target Site* above), the agent operates in an active mode and can suggest attack strategies to OpenText DAST to improve accuracy and performance. To use this feature, you must be using the Seven Pernicious Kingdoms taxonomy.

Application settings: Database

To access this feature, click **Edit > Application Settings** and then select **Database**.

Connection settings for scan/report storage

Select the database that will store OpenText DAST scan and report data. The choices are:

- **Use SQL Server Express** (for SQL Server Express Edition). Data for each scan will be stored in a separate database.
- **Use SQL Server** (for SQL Server Standard Edition). Data for multiple scans will be stored in a single database. 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.
- **Use PostgreSQL®**.

SQL Server database privileges

The account specified for the database connection must also be a database owner (DBO) for the named database. However, the account does not require sysadmin (SA) privileges for the database server. If the database administrator (DBA) did not generate the database for the specified user, then the account must also have the permission to create a database and to manipulate the security permissions. The DBA can rescind these permissions after OpenText DAST sets up the database, but the account must remain a DBO for that database.

Configuring SQL Server Standard Edition

To configure a profile for SQL Server Standard Edition:

1. Click **Configure**.
   The **Manage Database Settings** dialog box appears.
2. Click **Add**.
   The **Add Database** dialog box appears.
3. Enter a **Name** for the database profile.
4. Select a server from the **Server Name** list.

   Important! If SQL Server Browser is not enabled, the database server may not appear in the list. In this case, you must manually enter the connection information. The connection string is formatted as follows:

   ```
   SERVER\INSTANCE,PORT
   ```

Note that the port definition is added with a comma instead of a colon or semicolon.

5. In the **Log on to the server** group, specify the type of authentication used for the selected server:
   - **Use Windows Authentication** - Log on with your 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 dialog box.
8. Click **OK** to close the Manage Database Settings dialog box.

Configuring PostgreSQL

To configure a profile for PostgreSQL:

1. Select **Use PostgreSQL**.
2. Click **Configure**.
   The **Manage Database Settings** dialog box appears.
3. Click **Add**.
   The **Add Database** dialog box appears.
4. In the **Add Database** dialog box:
   - Enter a **Database Profile Name**.
   - Select a server from the **Server name** list.
   - In the Log on to the server group, specify the type of authentication used for the selected server:
     - **Use Windows Authentication** - Log on with your Windows account name and password.
     - **PostgreSQL Server Authentication** - use PostgreSQL server authentication.
   - Enter or select a specific database, or click **New** to create a database.
5. Click **OK** to close the Add Database dialog box.
6. Click **OK** to close the Manage Database Settings dialog box.

Connection settings for scan viewing

When displaying a list of scans (using either the Manage Scans view or the Report Generator wizard), OpenText DAST 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 Standard**: Select this option if you want to access data in SQL Server Standard Edition. See "Configuring SQL Server Standard Edition" on page 435 for instructions.
- **Show Scans Stored in PostgresSQL**: Select this option if you want to access data in PostgresSQL.

Application settings: Directories

To access this feature, click **Edit > Application Settings** and then select **Directories**.

Changing where OpenText DAST files are saved

You can change the locations in which OpenText DAST files are saved. To change locations:

1. Click the ellipsis button ... next to a category of information.
2. Use the Browse For Folder dialog box to select or create a directory.
3. Click **OK**.

Application settings: License

To access this feature, click **Edit > Application Settings** and then select **License**.

License details

This section provides pertinent information about the OpenText DAST license. If you want to change certain provisions of the license, click **Configure Licensing**, which will invoke the License Wizard.

The contents of the lower section of the window depend on the type of license management currently employed:

- Connected directly to the OpenText license server. See "Direct connection to OpenText" on the next page.
- Connected to a local OpenText™ Fortify License and Infrastructure Manager (LIM). See "Connection to LIM" on the next page.

Direct connection to OpenText

Options are described in the following table.

Option	Description
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	OpenText DAST 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 will contact the license server and release your license, allowing you to install OpenText DAST on another computer. 3. At the new computer, access the OpenText DAST application settings for licensing and enter the activation token.

Connection to LIM

Select the manner in which you want the LIM to handle the OpenText DAST license assigned to this computer. Options are described in the following table.

Option	Description
Connected License	The computer can run the Fortify software only when the computer is able to contact the LIM. Each time you start the 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 Fortify software anywhere, even when disconnected from your corporate intranet (on which the LIM is

Option	Description
	normally located), but only until the expiration date you specify. This enables you to take your laptop to a remote site and run the software. When you reconnect to the corporate intranet, you can access the Application License settings and reconfigure from Detached to Connected.

For more information about configuring OpenText DAST to use the LIM, see *OpenText™ Dynamic Application Security Testing Installation Guide*.

Application settings: Server Profiler

To access this feature, click **Edit > Application Settings** and then select **Server Profiler**.

Before starting a scan, OpenText DAST can invoke the Server Profiler to conduct a preliminary examination of the target website 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, click **Profile** (or select **Run Profiler Automatically**) on Step 4.

By default, 10 specific modules are enabled. To exclude a module, clear its associated check box.

Modules

The Server Profiler modules are described in the following table.

Module	Description
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 OpenText DAST'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, OpenText DAST would endlessly crawl these dynamic

Module	Description
	folders. This module determines if the site contains valid URLs that extend beyond that limit and, if so, enables you to increase the setting.
Verify client authentication protocol	This module determines which authentication (sign-in) protocol, if any, is required. OpenText DAST supports ADFS CBT, Automatic, Digest, HTTP Basic, Kerberos, and NTLM.
Check for additional hosts	This module searches the target site for references to additional host servers and enables 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 OpenText DAST 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 OpenText DAST from auditing non-essential responses.
Check for session state embedded in URLs	Instead of using cookies, some servers embed session state in URLs. OpenText DAST 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	OpenText DAST settings prevent pages with certain file extensions from being audited (see "Audit settings: Session Exclusions" on page 421). 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	An OpenText DAST scan setting specifies the maximum response size allowed; the default is 1,000 kilobytes. This module attempts to detect responses larger than the maximum

Module	Description
	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 OpenText DAST 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.
Check for cross-site request forgery	Cross-site request forgery, also known as a one-click attack or session riding, is often abbreviated as CSRF. CSRF is a type of website exploit where unauthorized commands are transmitted from a user that the website trusts. Unlike cross-site scripting , which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser. For more on CSRF, see "CSRF" on page 381.
Check for WebSphere servers	WebSphere servers require additional settings changes; enables the Profiler to detect these changes are required.

Application settings: Step Mode

To access this feature, click **Edit > Application Settings** and then select **Step Mode**.

Options for Step Mode are described in the following table.

Option	Description
Default Audit Mode	Select one of the following choices: • **Audit as you browse**: While you are navigating a target Web site, OpenText DAST concurrently audits the pages you visit. • **Manual Audit**: This option enables you to pause the Step Mode scan and return to OpenText DAST, where you can select a specific session and audit it.
Proxy Listener	Select the following options:

Option	Description
	• **Local IP Address**: Step Mode requires a proxy. Specify the IP address that the proxy should use. • **Port**: Specify the port that the proxy should use, or select **Automatically Assign Port**.

Application settings: Two-Factor Authentication

To access this feature, click **Edit > Application Settings** and then select **Two-Factor Authentication**.

Two-factor authentication control center

"Something you have" two-factor authentication involves an application server sending an SMS or email response to the user upon login to the web application. To use two-factor authentication in a scan, you must configure a Node.js server as a control center to process the SMS and email responses coming from your application server. For more information, see "Using two-factor authentication" on page 192.

To configure the control center:

1. In the **Local IP Address** drop-down list, select an IP address.

   Note: These IP addresses are available on the machine where OpenText DAST is installed.

2. Do one of the following:
   - To use a specific port, select the port from the **Port** list.
   - To have OpenText DAST choose the port, select the **Automatically Assign Port** check box.

     Important! The port for the control center must be exposed in the firewall so that the mobile application can access the server.

3. Click **Initialize**.

   The control center is started.

Mobile application

If your application server sends SMS responses, then you must install the **Fortify2FA** mobile application and download your two-factor authentication settings to it. After configuration, the mobile application receives the SMS response and forwards it to the control center.

Note: Currently, the mobile application is available only for Android operating systems.

To configure the mobile application:

1. In the **Mobile Phone Number** box, enter the phone number that will receive SMS responses.
2. Click **Generate QR Code**.

   The control center generates a quick response (QR) code that includes the two-factor authentication settings and a link to download the mobile application.
3. Install and configure the mobile application. For more information, see "Installing and configuring the Fortify2FA mobile app" below.

   Tip: If you use multiple threads in the scan, you might want to use more than one phone. Using the same phone number for multi-user scans can affect the scan time.
4. (Optional) To configure the mobile application for another phone, repeat steps 1-3.

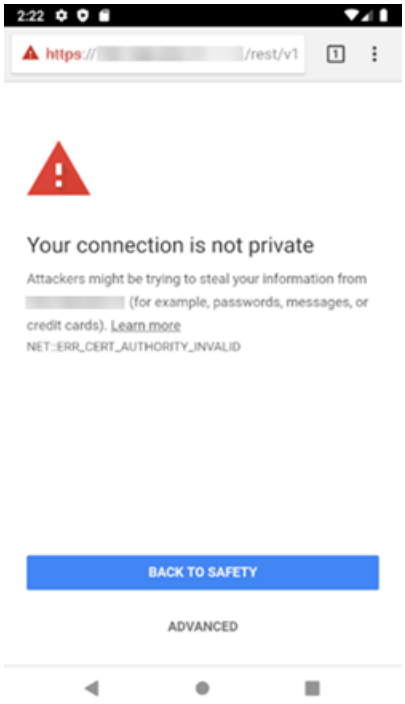
Installing and configuring the Fortify2FA mobile app

To install and configure the mobile application on the phone that will receive SMS responses:

1. Use the mobile phone's camera to scan the QR code in the **Two-factor Authentication Mobile Application** settings.

   A link appears.
2. Click the link (or **Open** button) to access the site for downloading the app.

A warning about the self-signed certificate appears.

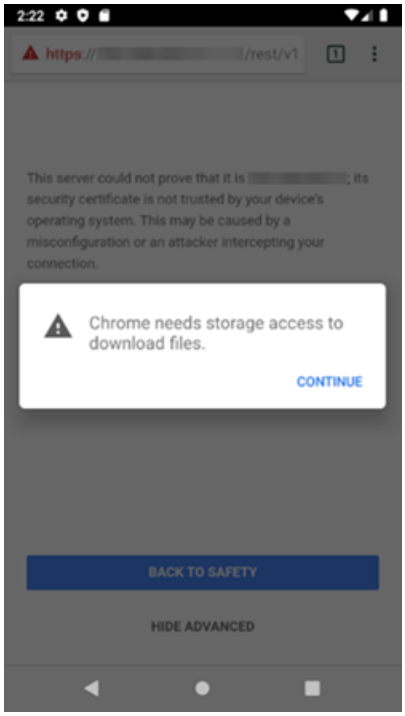


3. Click **ADVANCED**.
   Additional information is provided along with a link to proceed.

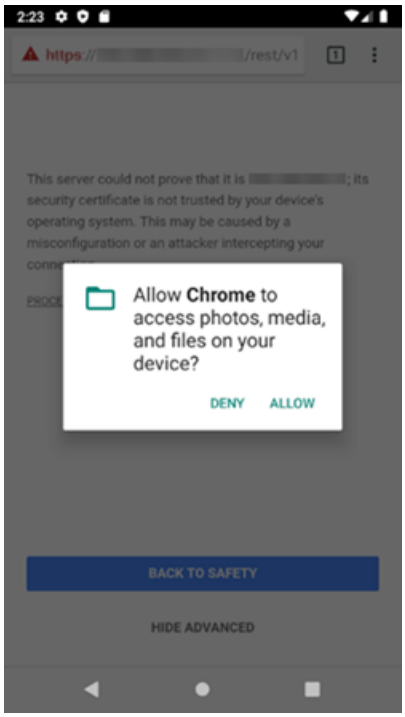


4. Click **PROCEED TO *<ip_address>* (UNSAFE)**.

A prompt requests storage access to download files.



5. Click **CONTINUE**.
   A prompt requests access to photos, media, and files on the device.

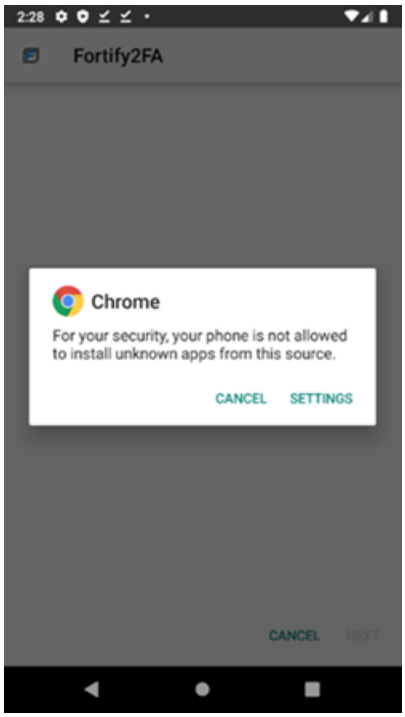


6. Click **ALLOW**.

The fortify-2fa.apk file is downloaded.

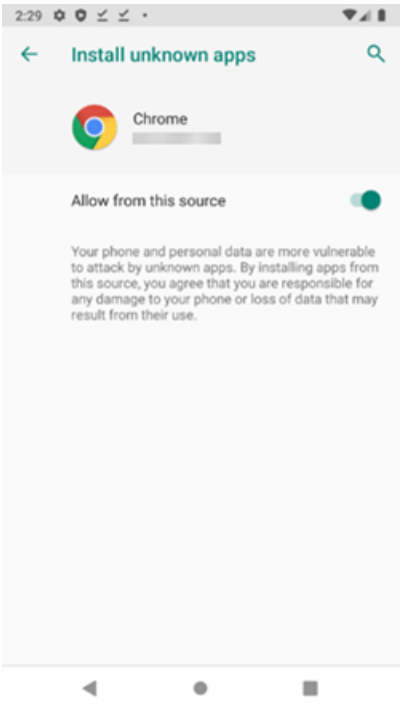


7. Click **OPEN**.
   A prompt advises about installing unknown apps.

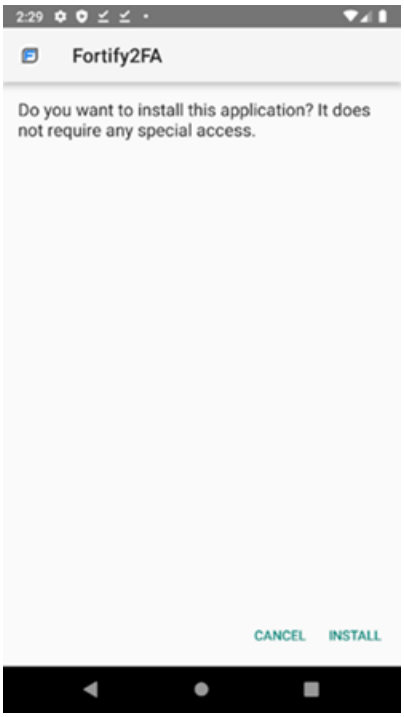


8. Click **SETTINGS**.

The Install unknown apps setting appears.

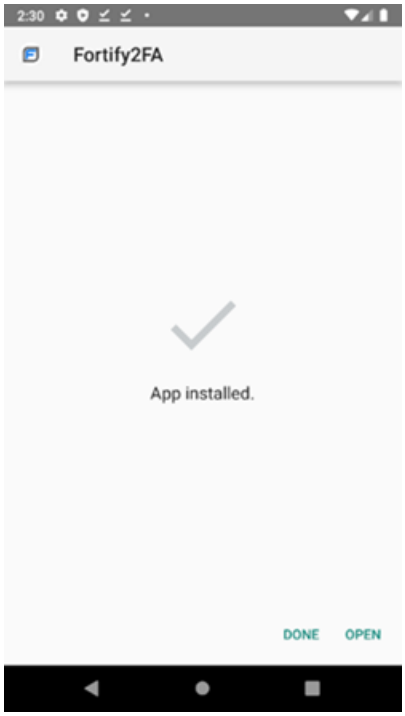


9. Enable **Allow from this source**.
   A prompt asks if you want to install the application.

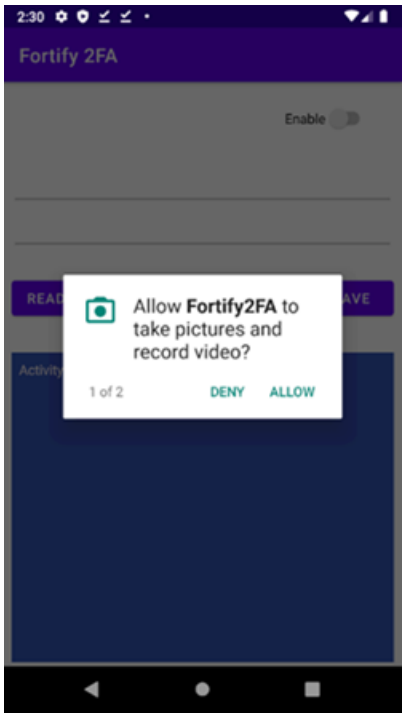


10. Click **INSTALL**.

A message indicates that the app is installed.

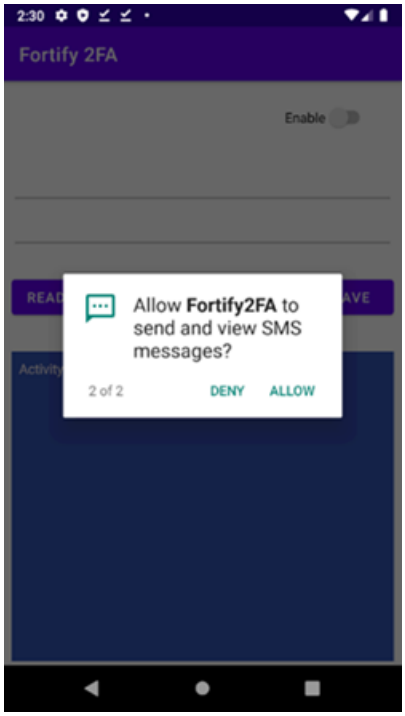


11. Click **OPEN**.
    A prompt requests permission to take pictures and record video.

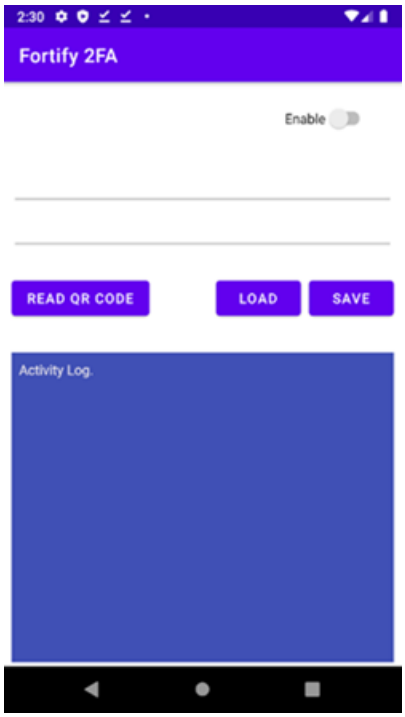


12. Click **ALLOW**.

A prompt requests permission to send and view SMS messages.



13. Click **ALLOW**.
    The app is ready to be configured.



14. Click **READ QR CODE** to scan the QR code in the **Two-factor Authentication Mobile Application** settings.

The two-factor authentication settings are configured in the **Fortify2FA** mobile application.

Application settings: Logging

To access this feature, click **Edit > Application Settings** and then select **Logging**.

The Logging options are described in the following table.

Option	Description
Clear Logs	Click this button to clear all logs.
Minimum Logging Level	Specify how OpenText DAST 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**, OpenText DAST 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. Alternatively, you can elect to **Never Purge** log files.
Rolling Log File Maximum Size	Specify the maximum size (in kilobytes) that any log file may attain. When a file reaches this limit, OpenText DAST simply stops writing to it.

Application settings: Proxy

To access this feature, click **Edit > Application Settings** and then select **Proxy Settings**.

OpenText DAST web services are used for update and support communications. Configure how these services are accessed in the Proxy Settings.

Not using a proxy server

If you are not using a proxy server to access these services, select **Direct Connection (proxy disabled)**.

Using a proxy server

If you are required to use a proxy server to access these services, select an option as described in the following table.

Option	Description
Auto detect proxy settings	Use the Web Proxy Autodiscovery (WPAD) protocol to locate a proxy autoconfig file and configure the browser's web proxy settings.
Use System Proxy settings	Import your proxy server information from the local machine. **Note:** Electing to use system proxy settings does not guarantee that you will access the Internet through a proxy server. If the Windows setting "Use a proxy server for your LAN" is not selected, then a proxy will not be used.
Use Firefox proxy settings	Import your proxy server information from Firefox. **Note:** Electing to use Firefox proxy settings does not guarantee that you will access the Internet through a proxy server. If the Firefox browser connection settings are configured for "No proxy," then a proxy will not be used.
Configure a proxy using a PAC file	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. See "Configuring a proxy" below in this topic.

Configuring a proxy

To configure a proxy:

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.

   Important! Smart Update is not available if you use a SOCKS4 or SOCKS5 proxy server configuration. Smart Update is available only when using a standard proxy server.

3. If authentication is required, select a type from the **Authentication** list. Options are:
   - **Automatic**

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

   - **Digest**
   - **HTTP Basic**
   - **NT LAN Manager (NTLM)**
   - **Kerberos**
   - **Negotiate**
4. If your proxy server requires authentication, enter the qualifying user name and password.

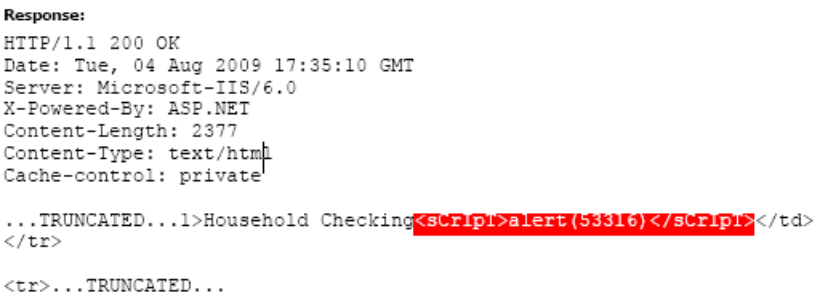
Application settings: Reports

To access this feature, click **Edit > Application Settings** and then select **Reports**.

Options

The Reports options are described in the following table.

Option	Description
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 OpenText DAST 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

Option	Description
	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 OpenText DAST Master Report uses the following 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.

Application settings: override SQL database settings

To access this feature, click **Edit > Application Settings > Run as a Sensor > Configure**.

Override database settings

OpenText DAST normally stores scan data in the device you specify in the Application Settings for database connectivity. For more information, see "Application settings: Database" on page 435.

Configure SQL Database

To configure SQL Database settings for OpenText DAST as a sensor:

1. On the Application Settings window, select **Override Database Settings**, and then click **Configure**.

   The Configure SQL Settings dialog box appears.
2. Select one of the following options:
   - **Use SQL Server Express**
   - **Use SQL Server**
3. If you selected **Use SQL Server Express**, click **OK** to complete the task and return to the Application Settings window.
4. If you selected **Use SQL Server**, then type the **Server Name** or select a Server Name from the list.
5. To update the server name, click **Refresh**.
6. In the Log on to the server area, select one of the following authentication options:
   - **Use Windows Authentication**
   - **Use SQL Server Authentication**
7. Type the **User name** and **Password** to log on to the server. In the Connect to a Database area, **Select or enter a database name** from the list, or click **New** to browse to a database.
8. Click **OK**.

Application settings: Smart Update

To access this feature, click **Edit > Application Settings** and then select **Smart Update**.

Options

The Smart Update options are described in the following table.

Option	Description
Language	Select a language to localize the security and report content in SecureBase. For more information, see "Selecting a different language" below.
Service	Enter the URL for the Smart Update service. The default is: https://smartupdate.fortify.microfocus.com/
Enable Smart Update on Startup	Select this option to check for updates automatically when starting OpenText DAST.

For more information, including instructions for updating OpenText DAST that is offline, see "SmartUpdate" on page 279.

Selecting a different language

When you select a different language for security and report content, OpenText DAST archives the original SecureBase content, creates a new SecureBase, and runs SmartUpdate to populate the new SecureBase with localized content. Afterwards, OpenText DAST must restart to use the new SecureBase instance.

To select a different language:

1. From the **Language** list, select the target language.
2. Click **OK**.

   A warning dialog box opens, prompting you to run SmartUpdate and advising you that OpenText DAST will restart to change to the new SecureBase instance.

   Note: After the blank SecureBase is populated with data, if you switch between languages and OpenText DAST detects that updated check data is available, you will be prompted to run SmartUpdate. Additionally, switching between languages requires OpenText DAST to restart.

3. Do one of the following:
   - To create a new SecureBase, populate it with localized content, and restart OpenText DAST, click **OK**.
   - To cancel the process and continue using the original SecureBase, click **Cancel**.

Application settings: Support Channel

To access this feature, click **Edit > Application Settings** and then select **Support Channel**.

The OpenText DAST support channel allows OpenText DAST to send data to and download messages from OpenText. It is used primarily for sending logs and "false positive" reports and for receiving "What's New" notices.

Opening the support channel

Select the **Allow connection to OpenText** option to open the OpenText DAST support channel. You may then specify the following:

- Support Channel URL - The default is:
  https://supportchannel.fortify.microfocus.com/service.asmx
- Download Directory - The default is:
  C:\ProgramData\HP\HP WebInspect\SupportChannel\Download\

Application settings: OpenText ALM

To access this feature, click **Edit > Application Settings** and then select **OpenText ALM**.

To integrate OpenText DAST with OpenText Application Lifecycle Management (ALM), you must create one or more profiles that describe the ALM server, project, defect priority, and other attributes. You can then convert an OpenText DAST vulnerability to an ALM defect and add it to the ALM database.

ALM License Usage

Creating or editing a profile consumes a license issued to ALM. The license is released, however, when the ALM application settings are closed. Similarly, sending a vulnerability to ALM consumes a license, but it is released after the vulnerability is sent.

Before You Begin

Make sure that the ALM Client Registration Add-in is installed on the same machine as OpenText DAST before creating a profile. Refer to your ALM documentation for more details.

Creating a Profile

To create a profile:

1. Click **Add**, and then enter a profile name in the Add Profile dialog box.
2. Enter or select the URL of an ALM server. If you haven't previously visited an ALM site, the list is empty. To enter a URL, use the format http://<*qc-server*>/qcbin/. Do not append "start_a.htm" (or other file name) to the URL.
3. Enter the user name and password that will allow you to access the server, and then click **Authenticate**.

   If the authentication credentials are accepted, the server populates the **Domain** and **Project** lists.
4. Click **Connect**, and then select a subject in the **Defect Reporting** group.
5. From the **Defect priority** list, select a priority that will be assigned to all OpenText DAST vulnerabilities reported to ALM using this profile.
6. Use the **Assign defects to** list to select the person to whom the defect will be assigned, and then select an entry from the **Project found in** list.
7. Use the remaining lists to map the OpenText DASTvulnerability rating to an ALM defect rating. If you select **Do Not Publish**, the vulnerability will not be exported. You must select at least one of the file mappings.
8. To export notes and screenshots associated with an OpenText DAST vulnerability, select **Upload vulnerability attachments to defect**.
9. 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, OpenText DAST prompts you to enter it.

Chapter 10: Reference lists

This chapter provides lists of OpenText DAST policies, scan log messages, and HTTP status codes.

OpenText DAST policies

A policy is a collection of vulnerability checks and attack methodologies that OpenText DAST deploys against a web application. Each policy is kept current through SmartUpdate functionality, ensuring that scans are accurate and capable of detecting the most recently discovered threats.

OpenText DAST contains the following packaged policies that you can use to determine the vulnerability of your web application.

Note: This list might not match the policies that you see in your product. SmartUpdate might have added or deprecated policies since this document was produced.

About OAST-related checks

For networks that have Internet access, OpenText DAST uses a public DNS service when running OAST-related checks. Ensure that your firewall does not block access to **fortify-oast.net**. For networks lacking Internet access, the Fortify OAST on Docker image is available. For more information, see the *OpenText™ Dynamic Application Security Testing and OAST on Docker User Guide*.

Best Practices

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

- **API**: This policy contains checks that target various issues relevant to an API security assessment. This includes various injection attacks, transport layer security, and privacy violation, but does not include checks to detect client-side issues and attack surface discovery such as directory enumeration or backup file search checks. All vulnerabilities detected by this policy may be directly targeted by an attacker. This policy is not intended for scanning applications that consume web APIs.
- **CWE Top 25 *<version>***: The Common Weakness Enumeration (CWE) Top 25 Most Dangerous Software Errors (CWE Top 25) is a list created by MITRE. The list demonstrates the most widespread and critical software weaknesses that can lead to vulnerabilities in software.

- **DISA STIG *<version>***: The Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) provides security guidance for use throughout the application development lifecycle. This policy contains a selection of checks to help the application meet the secure coding requirements of the DISA STIG *<version>*. Multiple versions of the DISA STIG policy may be available in the **Best Practices** group.
- **General Data Protection Regulation (GDPR)**: The EU General Data Protection Regulation (GDPR) replaces the Data Protection Directive 95/46/EC and provides a framework for organizations on how to handle personal data. The GDPR articles that pertain to application security and require businesses to protect personal data during design and development of their products and services are as follows:
  - Article 25, data protection by design and by default, which requires businesses to implement appropriate technical and organizational measures for ensuring that, by default, only personal data that is necessary for each specific purpose of the processing is processed.
  - Article 32, security of processing, which requires businesses to protect their systems and applications from accidental or unlawful destruction, loss, alteration, unauthorized disclosure of or access to personal data.

  This policy contains a selection of checks to help identify and protect personal data specifically related to application security for the GDPR.
- **NIST-SP80053R5**: NIST Special Publication 800-53 Revision 5 - (NIST SP 800-53 Rev.5) provides a list of security and privacy controls designed to protect federal organizations and information systems from security threats. This policy contains a selection of checks that must be audited to meet the guidelines and standards of NIST SP 800-53 Rev.5.
- **OWASP API Top 10 *<year>***: The OWASP API Top 10 *<year>* provides a list of the top security risks affecting APIs for the year specified. It aims to raise awareness around API security weaknesses and to educate those involved in API development and maintenance, such as developers, designers, architects, managers and/or organizations in general who need to secure web APIs. The OWASP API Top 10 focuses on weaknesses affecting web APIs and it is not intended to be used only by itself, instead it is intended to be used in combination with other standards and best practices to thoroughly capture all relevant risks. For example, it should be used in combination with the OWASP Top 10 to identify issues related to input validation such as injections.
- **OWASP Application Security Verification Standard (ASVS)**: The Application Security Verification Standard (ASVS) is a list of application security requirements or tests that can be used by architects, developers, testers, security professionals, tool vendors, and consumers to define, build, test, and verify secure applications.

  This policy uses OWASP ASVS suggested CWE mapping for each category of Securebase checks to include. Because CWE is a hierarchical taxonomy, this policy also includes checks that map to additional CWEs that are implied from OWASP ASVS suggested CWE using a "ParentOf" relationship.

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

By Type

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

- **Aggressive SQL Injection**: This policy performs a comprehensive security assessment of your web application for SQL Injection vulnerabilities. SQL Injection is an attack technique that takes advantage of non-validated input vulnerabilities to pass arbitrary SQL queries and/or commands through the web application for execution by a backend database. This policy performs a more accurate and decisive job, but has a longer scan time.
- **Apache Struts**: This policy detects supported known advisories against the Apache Struts framework.
- **Blank**: This policy is a template that you can use to build your own policy. It includes an automated crawl of the server and no vulnerability checks. Edit this policy to create custom policies that only scan for specific vulnerabilities.
- **Client-side**: This policy intends to detect all issues that require an attacker to perform phishing in order to deliver an attack. These issues are typically manifested on the client, thus enforcing the phishing requirement. This includes Reflected Cross-site Scripting and various HTML5 checks. This policy may be used in conjunction with the Server-side policy to provide coverage across both the client and the server.
- **Criticals and Highs**: Use the Criticals and Highs policy to quickly scan your web applications for the most urgent and pressing vulnerabilities while not endangering production servers. This policy checks for SQL Injection, Cross-Site Scripting, and other critical and high severity vulnerabilities. It does not contain checks that may write data to databases or create denial-of-service conditions, and is safe to run against

production servers.

- **Cross-Site Scripting**: This policy performs a security scan of your web application for cross-site scripting (XSS) vulnerabilities. XSS is an attack technique that forces a website to echo attacker-supplied executable code, such as HTML code or client-side script, which then loads in a user's browser. Such an attack can be used to bypass access controls or conduct phishing expeditions.
- **DISA STIG *<version>***: The Defense Information Systems Agency (DISA) Security Technical Implementation Guide (STIG) provides security guidance for use throughout the application development lifecycle. This policy contains a selection of checks to help the application meet the secure coding requirements of the DISA STIG *<version>*. Multiple versions of the DISA STIG policy may be available in the **By Type** group.
- **Mobile**: A mobile scan detects security flaws based on the communication observed between a mobile application and the supporting backend services.
- **NoSQL and Node.js**: This policy includes an automated crawl of the server and performs checks for known and unknown vulnerabilities targeting databases based on NoSQL, such as MongoDB, and server side infrastructures based on JavaScript, such as Node.js.
- **OAST**: This policy includes all checks that use Out-of-Band Application Security Testing technology in scanning logic.

  Note: For networks that have Internet access, OpenText DAST uses a public DNS service. For networks lacking Internet access, the Fortify OAST on Docker image is available. For more information, see the *OpenText™ Dynamic Application Security Testing and OAST on Docker User Guide*.

- **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.
- **PCI DSS 4.0**: The Payment Card Industry Data Security Standard 4.0 (PCI DSS 4.0) provides a baseline of technical and operational requirements designed to protect account data. This policy contains a selection of checks that need to be audited to meet the secure coding requirements of PCI DSS 4.0.
- **PCI Software Security Framework *<version>* (PCI SSF *<version>*)**: The PCI SSF provides a baseline of requirements and guidance for building secure payment systems and software that handle payment transactions. This policy contains a selection of checks that must be audited to meet the secure coding requirements of PCI SSF.
- **Privilege Escalation**: The Privilege Escalation policy scans your web application for programming errors or design flaws that allow an attacker to gain elevated access to data and applications. The policy uses checks that compare responses of identical requests with different privilege levels.
- **Server-side**: This policy contains checks that target various issues on the server-side of an application. This includes various injection attacks, transport layer security, and privacy violation, but does not include attack surface discovery such as directory

enumeration or backup file search. All vulnerabilities detected by this policy may be directly targeted by an attacker. This policy may be used in conjunction with the Client-side policy to provide coverage across both the client and the server.

- **SQL Injection**: The SQL Injection policy performs a security scan of your web application for SQL injection vulnerabilities. SQL injection is an attack technique that takes advantage of non-validated input vulnerabilities to pass arbitrary SQL queries and/or commands through the web application for execution by a backend database.
- **Transport Layer Security**: This policy performs a security assessment of your web application for insecure SSL/TLS configurations and critical transport layer security vulnerabilities, such as Heartbleed, Poodle, and SSL Renegotiation attacks.
- **WebSocket**: This policy detects vulnerabilities related to WebSocket implementation in your application.

Custom

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

Hazardous

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

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

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

Deprecated Checks and Policies

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

- **Aggressive Log4Shell (Deprecated)**: This policy performs a comprehensive security assessment of your web application for JNDI Reference injections in vulnerable versions of Apache Log4j libraries. In vulnerable versions, Log4j does not restrict JNDI features. This allows an attacker who can control log messages to inject JNDI references that point to an attacker-controlled server. This can lead to remote code execution on the vulnerable target. Compared with other policies that include

Log4Shell agent, this policy performs a more accurate and decisive job, but produces a significant number of requests and has a longer scan time.

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

vulnerabilities at the web server, web application server and web application layers. A quick scan does not run checks that are likely to create denial-of-service conditions, so it is safe to run on production systems.

- **Safe (Deprecated)**: A Safe scan includes an automated crawl of the server and performs checks for most known vulnerabilities in major packages and some unknown vulnerabilities at the web server, web application server and web application layers. A safe scan does not run any checks that could potentially trigger a denial-of-service condition, even on sensitive systems.
- **Standard (Deprecated)**: Standard (Deprecated) policy is copy of the original standard policy before it was revamped in R1 2015 release. A standard scan includes an automated crawl of the server and performs checks for known and unknown vulnerabilities at the web server, web application server and web application layers. A standard scan does not run checks that are likely to create denial-of-service conditions, so it is safe to run on production systems.

Scan log messages

This topic describes the messages that appear in the scan log. Messages are arranged alphabetically.

Note: For information about Alert-level messages in the scan log, see "Troubleshooting alerts" on page 493.

Audit Engine Initialization Error

Full Message

Audit Engine initialization error, engine:%engine%, error:%error%"

Description

An unrecoverable error occurred while attempting to initialize an audit engine. Contact Customer Support.

Argument Descriptions

Engine: The engine that was attempting to initialize.

Error: The actual error that occurred.

Possible Fixes

Not Applicable

External Links

Not Applicable

Auditor Error

Full Message

Error: Auditor error, session: <session ID> engine:<engine>, error:<error>

Description

An error occurred during an audit.

Argument Descriptions

Session: The session being audited when the error occurred.

Engine: The engine being run when the error occurred.

Error: The actual error that occurred.

Possible Fixes

Not Applicable

External Links

Not Applicable

Auto Response State Fail

Full Message

Auto Response State Fail detected. Please add response state rule.

Description

Automatic state detection has failed, but `Authorization: Bearer` was identified in requests.

Possible Fixes

If the token is a static token value, then ignore this alert.

If the token is dynamic, then create a response state rule. For more information, see "Scan settings: HTTP Parsing" on page 376.

External Links

Not Applicable

Check Error

Full Message

Error: Check error, session:8BE3AFEC5051507168B66AEC59C8915B, Check:10346, engine: SPI.Scanners.Web.Audit.Engines.RequestModify

Description

An error occurred while processing a check.

Argument Descriptions

Session: Session where the check error occurred.

Check: The check that encountered the problem.

Engine: The engine being run when the error occurred.

Error: The error.

Possible Fixes

Install the latest version of SmartUpdate.

External Links

Not Applicable

Completed Post-Scan Analysis Module

Full Message

Completed Post-Scan Analysis Module: %module%

Description

One of the post-scan analysis modules has ended.

Argument Descriptions

module: the name of the post-scan analysis module.

Possible Fixes

Not Applicable

External Links

Not Applicable

Concurrent Crawl and Audit Start

Full Message

Info:Concurrent Crawl and Audit Start

Description

This message indicates that Concurrent Crawl and Audit has started.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Concurrent Crawl and Audit Stop

Full Message

Info:Concurrent Crawl and Audit Stop

Description

This message indicates that Concurrent Crawl and Audit has stopped.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Concurrent Crawl Start

Full Message

Info:Concurrent Crawl Start:

Description

This message indicates that Concurrent Crawl has started.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Concurrent Crawl Stop

Full Message

Info:Concurrent Crawl Stop

Description

This message indicates that Concurrent Crawl has stopped.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Connectivity Issue, Reason

Full Message

Connectivity issue, Reason: FirstRequestFailed, HTTP Status:404,

Description This message indicates a network connectivity issue. OpenText DAST was unable to communication with the remote host.

Argument Descriptions

Reason: FirstRequestFailed - a requested has failed.
HTTP Status: 404 - The status returned for the failed request.

Possible Fixes

- Power cycle your network hardware

  If the issue persists, unplug your modem and router, wait a few seconds, then plug them back in. Sometimes, these devices simply need to be refreshed. This could be due to a network outage or improperly configured network settings.
- Use Microsoft's network diagnostic tools

  Open Network Diagnostics by right-clicking the network icon in the notification area, and then clicking Diagnose and repair.
- Check wiring

  Make sure that all wires are connected properly.
- Check host's power

  If you're trying to connect to another computer, make sure that computer is powered on.
- Check connection settings

  If the problem began after you installed new software, check your connection settings to see if they have been changed. Open Network Connections by clicking the Start button , clicking Control Panel, clicking Network and Internet, clicking Network and Sharing Center, and then clicking Manage network connections. Right-click the connection, and then click Properties. If you are prompted for an administrator password or confirmation, type the password or provide confirmation.
- Troubleshoot all Firewalls

External Links

Troubleshoot network connection problems

Internet Connectivity Evaluation Tool

Connectivity Issue, Reason, Error

Full Message

Connectivity issue, Reason:FirstRequestFailed, Error:Server:zero.webappsecurity.com:80, Error:(11001)Unable to connect to remote host : No such host is known:

Description

This message indicates a network connectivity issue. OpenText DAST was unable to communication with the remote host.

Argument Descriptions

Reason: FirstRequestFailed - a requested has failed.

Server: The server to which the request was sent.

Error: (11001)Unable to connect to remote host : No such host is known: - Communication to the remote host failed due to connectivity issues.

Possible Fixes

- Power cycle your network hardware

  If the issue persists, unplug your modem and router, wait a few seconds, then plug them back in. Sometimes, these devices simply need to be refreshed. This could be due to a network outage or improperly configured network settings.
- Use Microsoft's network diagnostic tools

  Open Network Diagnostics by right-clicking the network icon in the notification area, and then clicking Diagnose and repair.
- Check wiring

  Make sure that all wires are connected properly.
- Check host's power

  If you're trying to connect to another computer, make sure that computer is powered on.
- Check connection settings

  If the problem began after you installed new software, check your connection settings to see if they have been changed. Open Network Connections by clicking the Start button , clicking Control Panel, clicking Network and Internet, clicking Network and Sharing Center, and then clicking Manage network connections. Right-click the connection, and then click Properties. If you are prompted for an

administrator password or confirmation, type the password or provide confirmation.
- Troubleshoot all firewalls

External Links

Troubleshoot network connection problems

Internet Connectivity Evaluation Tool

Crawler Error

Full Message

Error: Crawler error, session: <session ID> error:<error>

Description

The crawler failed to process the session. Not user-correctable. Contact Customer Support.

Argument Descriptions

Session: The session in which the error occurred.

Error: The actual error.

Possible Fixes

Not Applicable

External Links

Not Applicable

Database Connectivity Issue

Full Message

Error: SPI.Scanners.Web.Framework.Session in updateExisting,retries failed, giving up calling iDbConnetivityHandler.OnConnectivityIssueDetected

Description

This message indicates that the database stopped responding.

Argument Descriptions

Error Text: Contains a description of the error that triggered the message

Possible Fixes

Make sure the database server is running and responding.

External Links

Not Applicable

Engine Driven Audit Start

Full Message

Info:Engine Driven Audit Start

Description

This message indicates Engine Driven Audit has started.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Engine Driven Audit Stop

Full Message

Info:Engine Driven Audit Stop

Description

This message indicates Engine Driven Audit has stopped.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Engine Driven Engine Start

Full Message

Info:Engine Driven Engine Start, Engine: LFI Agent

Description

This message indicates the engine indicated has started execution.

Argument Descriptions

Engine: The Engine that is starting.

Possible Fixes

Not Applicable

External Links

Not Applicable

Engine Driven Engine Stop

Full Message

Info:Engine Driven Engine Stop, Engine: LFI Agent Sessions Processed:406

Description

Engine driven audit completed for the specified engine.

Argument Descriptions

Engine: The Engine that has been stopped.

Sessions processed: Number of sessions processed by the engine.

Possible Fixes

Not Applicable

External Links

Not Applicable

External Correlation Enabled

Full Message

External Correlation Enabled, Origin:*<product_name>* OriginID:*<numeric_value>* OriginDateTime:*<date_time>* File:*<filename>*.json

Description

External correlation was automatically enabled for the scan.

Argument Descriptions

Origin: The external product that correlates, such as Fortify_SAST.

OriginID: The ID for the external scan that contains findings.

OriginDateTime: When the external scan was produced.

File: The JSON file that contains the external findings.

Possible Fixes

Not Applicable

External Links

Not Applicable

External Finding

Full Message

External Finding, Origin:*<product_name>* OriginID:*<numeric_value>* OriginDateTime:*<date_time>* OriginFindingID:*<guid>* FindingType:*<type>*

Description

Provides information about the finding in the external scan.

Argument Descriptions

Origin: The external product that correlates, such as Fortify_SAST.

OriginID: The ID for the external scan that contains findings.

OriginDateTime: When the external scan was produced.

OriginFindingID: The unique identifier of the finding in the external scan file.

FindingType: The type of finding, such as XSS, in the external scan file.

Possible Fixes

Not Applicable

External Links

Not Applicable

Finding Correlated

Full Message

Finding Correlated, Check:*<check_id><check_name>* Param:*<parameter_name>* Request:*<http_method><resource_url>*

Description

This finding correlates to a finding in the external scan.

Argument Descriptions

Check: The check ID and check name from SecureBase.

Param: The parameter name used in the attack.

Request: The HTTP request method (such as POST, PUT, GET) and the URL of the resource attacked.

Possible Fixes

Not Applicable

External Links

Not Applicable

License Issue

Full Message

Error: License issue: License Deactivated

Description

A problem has occurred with the license.

Argument Descriptions

Issue: The issue that occurred.

Possible Fixes

Make sure OpenText DAST is properly licensed.

External Links

Not Applicable

Log Message Occurred

Full Message :

<Level>: <ScanID> , <Logger>: <Exception>

Description:

Generic message for exceptions

Argument Descriptions

ScanID: Scan ID.

Logger: Name of logger.

Exception: The exception thrown.

Possible Fixes

Not Applicable

External Links

Not Applicable

Memory Limit Reached

Full Message

Warn: Memory limit reached: level:1,limit:1073610752, actual:1076625408.

Error: Memory limit reached: level:0,limit:1073610752, actual:1076625408.

Description

The memory limits of the WI process have been reached.

Argument Descriptions

Level: The severity of the problem.

Limit: The memory limit of the process.

Actual: The actual memory allocated to the process.

Possible Fixes

Close other scans that are not running.

Run only one scan at a time in a given OpenText DAST instance.

External Links

Not Applicable

Missing Session for Vulnerability

Full Message

Info: Missing Session for Vulnerability

Description

Cannot find session that is associated with a vulnerability.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

New Blind SQL Check Not Enabled

Full Message

New Blind SQL check (checkid newcheckid%) is not enabled. A policy with both check %newcheckid% and check %oldcheckid% enabled is recommended.

Description

The newer check for blind SQL injection is not included in the scan policy.

Argument Descriptions

newcheckid: The identifier of the newer SQL injection check (10962)

oldcheckid: The identifier of the older SQL injection check (5659)

Possible Fixes

Add the newer check (10962) to the scan policy.

External Links

Not Applicable

Persistent Cross-Site Scripting Audit Start

Full Message

Info:Persistent Cross-Site Scripting Audit Start

Description

Persistent Cross-Site Scripting Audit has started.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Persistent Cross-Site Scripting Audit Stop

Full Message

Info:Persistent Cross-Site Scripting Audit Stop

Description

Persistent Cross-Site Scripting Audit has stopped.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Post-Scan Analysis Started

Full Message

Post-Scan Analysis started.

Description

Post-scan analysis has begun. Additional messages will be displayed for each module used (authentication, macro, file not found, etc.).

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Post-Scan Analysis Completed

Full Message

Post-Scan Analysis completed.

Description

Post-scan analysis has ended. Additional messages will be displayed for each module used (authentication, macro, file not found, etc.).

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Reflect Audit Start

Full Message

Info:Reflect Audit Start

Description

Reflection phase started.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Reflect Audit Stop

Full Message

Info:Reflect Audit Stop

Description

Reflection phase completed.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Response State Rules Fail

Full Message

Response State Rules Fail detected for %count% rule(s). Name of rule(s): %ruleslist%.

Description

A response state rule is configured, but was not triggered during the scan.

Argument Descriptions

count: The number of rules that failed

ruleslist: The names of the rules that failed

Possible Fixes

Correct the regular expression in the rule or delete the rule. For more information, see "Scan settings: HTTP Parsing" on page 376.

External Links

Not Applicable

Scan Complete

Full Message

Info:Scan Complete, ScanID:<id-number>

Description

This message indicates that the scan has completed successfully.

Argument Descriptions

ScanID: Unique identifier of a scan

Possible Fixes

Not Applicable

External Links

Not Applicable

Scan Failed

Full Message

Info:Scan Failed, ScanID::<id-number>

Description

This message indicates that the scan has failed.

Argument Descriptions

ScanID: Unique identifier of a scan

Possible Fixes

Depends upon the reason the scan failed, which is specified in a different message.

External Links

Not Applicable

Scan Start

Full Message

Info:Scan Start, ScanID:<id-number> Version:X.X.X.X, Location:C:\Program Files\Fortify\Fortify WebInspect\WebInspect.exe

Description

This message indicates the start of a scan.

Argument Descriptions

ScanID: Unique identifier of a scan.

Version: Version of OpenText DAST running the scan.

Location: The physical location of the OpenText DAST executable.

Possible Fixes

Not Applicable

External Links

Not Applicable

Scan Start Error

Full Message

Scan start error: %error%

Description

An unrecoverable error occurred while starting the scan. Contact Customer Support.

Argument Descriptions

error: description of the problem.

Possible Fixes

Not Applicable

External Links

Not Applicable

Scan Stop

Full Message

Info:Scan Stop, ScanID:<id-number>

Description

This message indicates that the scan has been stopped.

Argument Descriptions

ScanID: Unique identifier of a scan.

Possible Fixes

Not Applicable

External Links

Not Applicable

Scanner Retry Start

Full Message

Info:Scanner Retry Start

Description

Retry phase started.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Scanner Retry Stop

Full Message

Info:Scanner Retry Stop

Description

Retry phase stopped.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Sequential Audit Start

Full Message

Info:Sequential Audit Start

Description

This message indicates that the Sequential Audit has started.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Sequential Audit Stop

Full Message

Info:Sequential Audit Stop

Description

This message indicates that the Sequential Audit has stopped.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Sequential Crawl Start

Full Message

Info:Sequential Crawl Start

Description

This message indicates that Sequential Crawl has started.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Sequential Crawl Stop

Full Message

Info:Sequential Crawl Stop

Description

This message indicates that the Sequential Crawl has stopped.

Argument Descriptions

Not applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Settings Override

Full Message

Settings Override, Setting:<setting, Original Value:<original>, New Value:<newValue>, Reason:<reason>

Description

A setting was changed by the product. This may indicate a setting upgrade issue.

Argument Descriptions

Setting: The setting that is being overridden.

Original Value: The original value of the setting.

New Value: The value to which the setting is being changed.

Reason: The reason for the override.

Possible Fixes

Restore factory defaults and reapply custom settings.

External Links

Not Applicable

SPA Frameworks Detected

Full Message

The crawl identified the following Single Page Application frameworks: %frameworks%. SPA support enabled.

Description

The crawler detected one or more Single Page Application (SPA) frameworks and enabled SPA support for the scan.

Argument Descriptions

frameworks: A list of the detected frameworks.

Possible Fixes

Not Applicable

External Links

Not Applicable

Start URL Error

Full Message

Start Url Error:%url%, error:%error%

Description

An unrecoverable error occurred processing the start URL. Check url syntax; if correct, contact Customer Support.

Argument Descriptions

url: The URL that caused the error.

error: Description of the error.

Possible Fixes

Not Applicable

External Links

Not Applicable

Start URL Rejected

Full Message

Start Url Rejected:%url%, reason:%reasons%, session:%session%

Description

The URL was rejected due to request rejection settings; settings should be modified or a different start URL used.

Argument Descriptions

url: the start URL

reason: Reason for the rejection.

session: The session during which the error occurred.

Possible Fixes

Not Applicable

External Links

Not Applicable

Starting Post-Scan Analysis Module

Full Message

Starting Post-Scan Analysis Module: %module%

Description

One of the post-scan analysis modules has begun.

Argument Descriptions

module: the name of the post-scan analysis module.

Possible Fixes

Not Applicable

External Links

Not Applicable

Stop Requested

Full Message

Info:Stop Requested, reason=Pause button pushed

Description

Scan is entering suspended state.

Argument Descriptions

Reason: Reason for the stop.

Possible Fixes

Not Applicable

External Links

Not Applicable

Verify Audit Start

Full Message

Info:Verify Audit Start

Description

Verify phase started.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Verify Audit Stop

Full Message

Info:Verify Audit Stop

Description

Verify phase completed.

Argument Descriptions

Not Applicable

Possible Fixes

Not Applicable

External Links

Not Applicable

Web Macro Error

Full Message

Error: Web Macro Error, Name: Login webmacro Error: RequestAborted

Description

An error occurred during playback of a web macro.

Argument Descriptions

Name: Name of the macro being played when the error occurred.

Error: The error that occurred.

Possible Fixes

Depends on the error encountered. For RequestAborted error, the server did not respond during macro playback. If this occurs frequently, the value of Request timeout should be increased. See Connectivity issue for other potential solutions.

External Links

Not Applicable

Web Macro Status

Full Message

Error: Web Macro Status, Name: login.webmacro Expected:302, Actual:200, Url:<URL>

Description

OpenText DAST received a response during macro playback that did not match the response obtained during the recording of the macro.

Argument Descriptions

Name: Name of the web macro.

Expected: The status code expected to be returned.

Actual: The status code that was actually returned.

URL: The target URL of the request.

Possible Fixes

This could indicate that OpenText DAST is attempting to log in when it is already logged in or that OpenText DAST is failing to log in. Check to see if OpenText DAST is successfully logged in during a scan. If not, record the login macro again.

External Links

Not Applicable

HTTP status codes

The following list of status codes was extracted from the Hypertext Transfer Protocol version 1.1 standard (RFC 2616). You can find more information at http://www.w3.org/Protocols/.

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-	The returned metainformation in the entity-header is not the

Code	Definition
Authoritative Information	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.
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

Code	Definition
	include a WWW-Authenticate header field (section 14.47) 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

Code	Definition
	of the request is in a format not supported by the requested resource for the requested method.
416 Requested Range Not Satisfiable	A server should return a response with this status code if a request included a Range request-header field (section 14.35), 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 (see section 14.20) 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.

Chapter 11: Troubleshooting

This chapter provides troubleshooting tables, information about testing login macros, and options for uninstalling OpenText DAST.

Troubleshooting OpenText DAST

The following paragraphs provide troubleshooting information for OpenText DAST and the OpenText DAST tools.

Connectivity issues

The following table describes issues with connectivity.

Symptom or Error Message	Possible Cause	Possible Solution
When using a macro recorder or the Guided Scan Wizard while testing a site that uses HTTPS rather than HTTP, there is no connectivity to the site.	The user running OpenText DAST does not have required access to the Windows MachineKeys folder.	Modify the permissions of C:\ProgramData\Microsoft\Crypto\RSA\MachineKeys. On the folder properties **Security** tab, use the **Advanced** button and configure permissions to allow full control for the user for **This folder, subfolders and files**.
The **Use OpenSSL Engine** application setting is selected and you see an error that contains the following text in the Guided Scan browser, in the Profiler results, or in the scan logs:	An OpenSSL defect has caused a connectivity issue with the target web application.	Ensure that the certificate or certificates that you use to connect to the target web application are marked as exportable. For more information, refer to your Windows documentation.

Symptom or Error Message	Possible Cause	Possible Solution
"Make this client Certificate key exportable in Certificate storage."		

Scan initialization failures

The following table describes issues with scan initialization.

Symptom or Error Message	Possible Cause	Possible Solution
Scan initialization fails when using SQL Express as the scan database.	The SQL Express service is not running.	Verify that the service is running. The service name is “SQL Server (SQLEXPRESS)" or similar.
	The SQL Express cache may have become corrupted.	To clear the cache: 1. Stop all SQL related services and processes. 2. Delete the SQL Express cache folder. A typical location is as follows or similar: C:\Users*<username>*\AppData\Local\Microsoft\Microsoft SQL Server Data\SQLEXPRESS 3. Restart the machine.
Scan initialization fails with errors related to loading SPI.Parsers.Script.	Windows may have failed to apply the Microsoft Visual C++ redistributable package for Visual Studio 2015, 2017, or 2019.	Manually install the C++ redistributable package before continuing.

Scan configuration issues

The following table describes issues that may occur while configuring a scan.

Symptom or Error Message	Possible Cause	Possible Solution
In Guided Scan, a truclientbrowser.exe system error occurs related to a missing .dll file.	Windows may have failed to apply the Microsoft Visual C++ redistributable package for Visual Studio 2015, 2017, or 2019.	Manually install the C++ redistributable package before continuing.

Troubleshooting alerts

Alerts do not always indicate that there is a scan quality issue. Some alerts may be false positive. However, alerts may provide insight into issues that could adversely affect the scan.

Disabling alerts

The alerts feature includes sample intervals and active intervals. Sample interval alerts may occur as often as once per minute in your scan log. Although these alerts may not indicate a functional issue with the scan, if the number of alerts received becomes problematic, contact Customer Support for assistance in disabling individual alerts or the alerts feature. For more information, see "Preface" on page 24.

Alerts troubleshooting table

Important! Any solutions involving changes to scan settings must be made for a future scan. You cannot change the scan settings for the current scan.

The following table describes possible causes and solutions for alerts.

Alert	Possible Cause	Possible Solution
Excessive logins detected	The login macro has been played an excessive number of times for the number of requests made. The login credentials may	Do one of the following: • Perform troubleshooting procedures on the macro. • Record a new login macro.

Alert	Possible Cause	Possible Solution
	be incorrect or the logout signature may be invalid.	For more information, see the *OpenText™ Dynamic Application Security Testing Tools Guide*.
Redundant content detected	Redundant content has been detected.	You might be able to improve performance by enabling redundant page detection. For more information, see "Scan settings: General" on page 361.
Excessive response time	Responses coming from the web server are taking longer than average or longer than expected. A longer response time may result in a slower scan.	Check your network connectivity or the performance of the application under test (AUT).
WAF detected	A web application firewall (WAF) signature has been detected.	Disable the WAF that is protecting the AUT.

Testing login macros

OpenText DAST performs tests on the login macro in the following instances:

- When an auto-generated macro, newly-recorded macro, or pre-existing macro is tested during scan configuration
- At the start of the scan with any login macro if **Enable macro validation** is selected in Scan Settings: Authentication

Note: Macro testing is not supported for macros containing two-factor authentication.

Validation tests performed

The following table describes the tests that OpenText DAST performs.

Test	Result of Failure
Determine if the validation step is missing.	The scan continues, but a warning is written to the scan log.
Monitor the behavior when incorrect credentials are used.	
Determine whether the landing page is accessible with a login state.	
Determine whether the landing page is accessible without a login state.	
Determine if the site can handle multiple logins concurrently. The default number of concurrent logins tested is 5.	
Verify that the auto-generated macro logs into the application.	The scan stops and an error is written to the scan log.
Verify that the replay of the macro logs into the application.	

If a scan stops after failing a test, it may be possible to examine the specific error message in the scan log to determine and resolve the issue. Use the error message and the troubleshooting tips in this topic to help resolve the issue.

Troubleshooting tips

In all cases of macro failure, it is possible that an invalid macro was recorded. However, a previously good macro that fails is almost always due to site changes or credentials.

The following table provides possible causes and solutions for each error message.

Note: This table does not include all possible causes and solutions for each error message. Additional troubleshooting may be necessary.

Error Message	Possible Cause	Possible Solution
Automatic login generation	The login macro could not	Try the Auto-gen Login

Error Message	Possible Cause	Possible Solution
failed	be created because the user credentials provided are not valid.	Macro option again using credentials that are known to be valid.
Execution Failed	An HTML element, such as a verification element, username, or password, was not located.	Record a new macro in the Web Macro Recorder to identify the login input elements.
	The username has been deactivated (removed from the database) and/or the password has changed.	Record a new macro in the Web Macro Recorder using credentials that are known to be valid.
Logged in verification step not found	The login macro does not contain a verification step.	Edit the macro in the Web Macro Recorder to add a verification step to indicate a successful login.
Verification step did not fail after invalid login	The verification step succeeded after an invalid login attempt. A valid verification step should only succeed upon successful login. This indicates that an incorrect login verification object was selected.	Edit the macro in the Web Macro Recorder to select another object for the verification step.

For specific information about using the Web Macro Recorder, see the *OpenText™ Dynamic Application Security Testing Tools Guide*.

Uninstalling OpenText DAST

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

Options for removing

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

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

Send Documentation Feedback

If you have comments about this document, you can contact the documentation team by email.

Note: If you are experiencing a technical issue with our product, do not email the documentation team. Instead, contact Customer Support at https://www.microfocus.com/support so they can assist you.

If an email client is configured on this computer, click the link above to contact the documentation team and an email window opens with the following information in the subject line:

Feedback on User Guide (Dynamic Application Security Testing 26.4.0)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to fortifydocteam@opentext.com.

We appreciate your feedback!