AppSpider Pro

User's Guide

Additional information on AppSpider can be found on our community page located at:
https://community.rapid7.com/community/appspider

Also see: https://help.rapid7.com

AppSpider Pro User Guide

Contents

AppSpider System Requirements

Server hardware requirements

- multi-core CPU
- 4 GB RAM
- 1 network interface card (NIC)
- 500 GB hard drive space

Software requirements

One of the following operating systems:

- Microsoft Windows Server 2008 Editions x32/x64 bit
- Microsoft Windows Server 2008 R2 Editions x32/x64 bit
- Microsoft Windows Vista Ultimate, Professional, Enterprise, Home Premium x32/x64 bit
- Microsoft Windows 7 Ultimate, Professional, Enterprise, Home Premium x32/x64 bit
- Microsoft Windows 7 Ultimate, Professional, Enterprise, Home Premium SP1 x32/x64 bit
- Microsoft Windows 8 Professional x64 bit
- Microsoft Windows 10

Additional requirements

- Microsoft .NET Framework 4.5 Redistributable Package.
- To successfully install AppSpider, you will need more free hard disk space than the size of the installer itself. For more specific information about the space requirements, please contact support@rapid7.com.

Installation

Installation instructions

Installing AppSpider Pro (hereafter referred to as AppSpider) involves a series of steps which must be completed in the correct sequence for the installation to be successful.

The installer is available on the customer site: https://www.rapid7.com/products/appspider/download.jsp.

Note: Most of the steps in this guide refer to the AppSpider graphical user interface. To install this component, select the checkbox next to **AppSpider 6 GUI**.

To install AppSpider:

1. Download the setup program and save the file to your hard disk.
2. Double-click the *AppSpider_Setup-xxx_xxx_xxx.exe* program file on your hard disk to start the setup program.

   Note: Installation should be run under Administrator account (Run as Administrator).
3. Follow the instructions on the screen to complete the installation.

Uninstall

1. To remove the download file, delete *AppSpider_Setup-xxx_xxx_xxx.exe.*
2. The default installation directory is C:\Program Files\Rapid7\AppSpider.
3. To remove the installed files, use AppSpider uninstaller (*uninst_AppSpider_6.exe*).

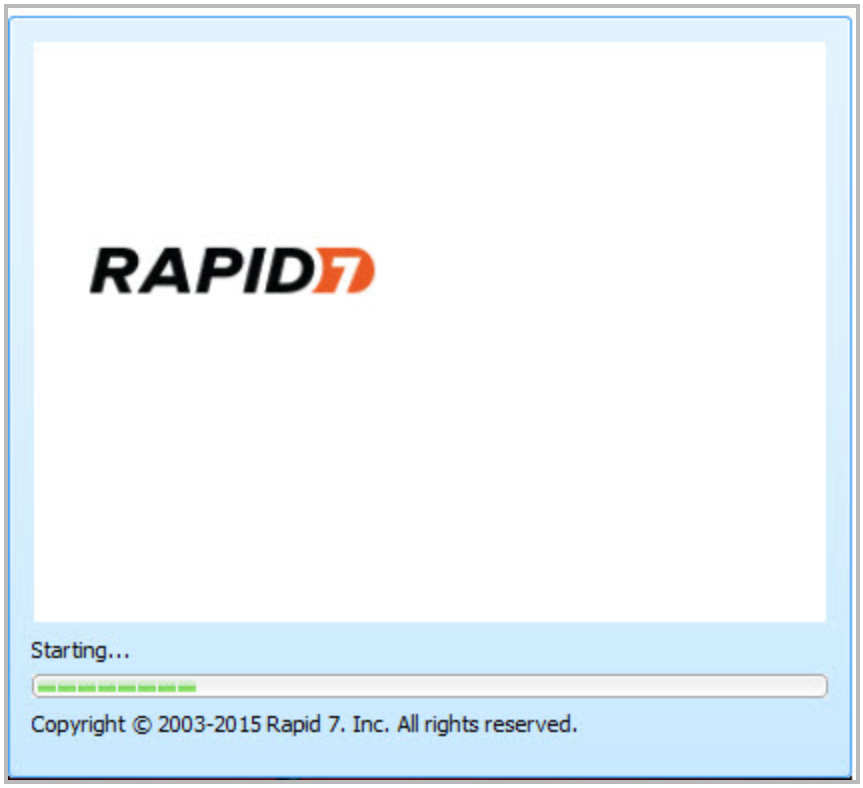
Start the AppSpider Pro User Interface

To use the AppSpider Graphical User Interface, it needs to have been selected during the installation process on the “Choose Components” window. For more information, see *Installation instructions* on page 6 .

To start the program, do one of the following:

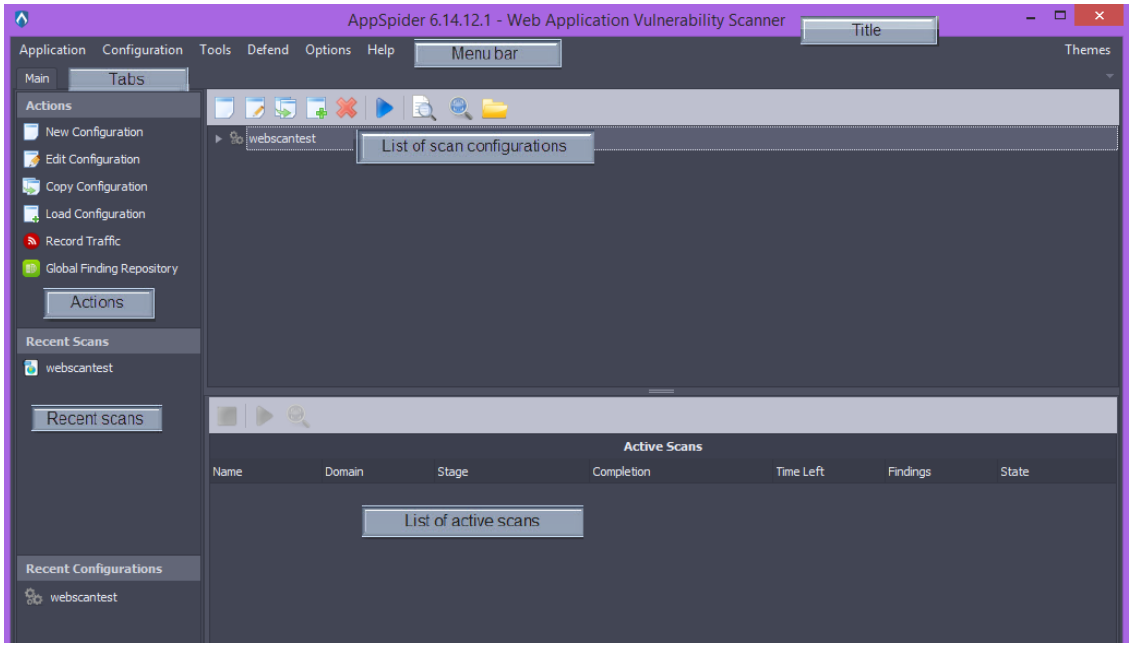
- Select the AppSpider shortcut on the desktop.
- Select Start -> All programs -> Rapid7 -> AppSpider

A splash screen will appear while the application is loading.



Main window

AppSpider's Main window contains the following information:



- *Title*: contains the application name and its version.
- *Tabs*: displays main, active scan and scan status tabs if they are opened.
- *Menu bar*: contains the following items: *Application*, *Configuration*, *Tools*, *Options*, *Help* and *Themes*. For more information, see *Menu bar items* on page 9.
- *Actions panel*: allows you to create or modify a scan configuration. For more information, see *Actions Panel* on page 10.

Menu bar items

Application

- **Exit** - closes the application.

Configuration

- **New** - opens **Scan Config - New** page for creating a new scan configuration
- **Load** - loads a selected scan configuration

Tools

- **Traffic recorder** - opens Traffic Recorder tab
- **Traffic viewer** - opens Traffic Viewer tab
- **RegEx builder** - opens RegEx builder window
- **Request builder** - opens Request builder tab
- **Attack policy** - opens Attack policy tab
- **Global findings repository** - opens the findings repository tab
- **Encode/Decode utility** - Opens the encode/decode utility tab
- **Swagger utility** - Opens the swagger utility tab (see page 90 for additional information)

Defend

- **Defend**- opens Defend UI (see page 83 for additoinal details on the Defend functionality).

Options

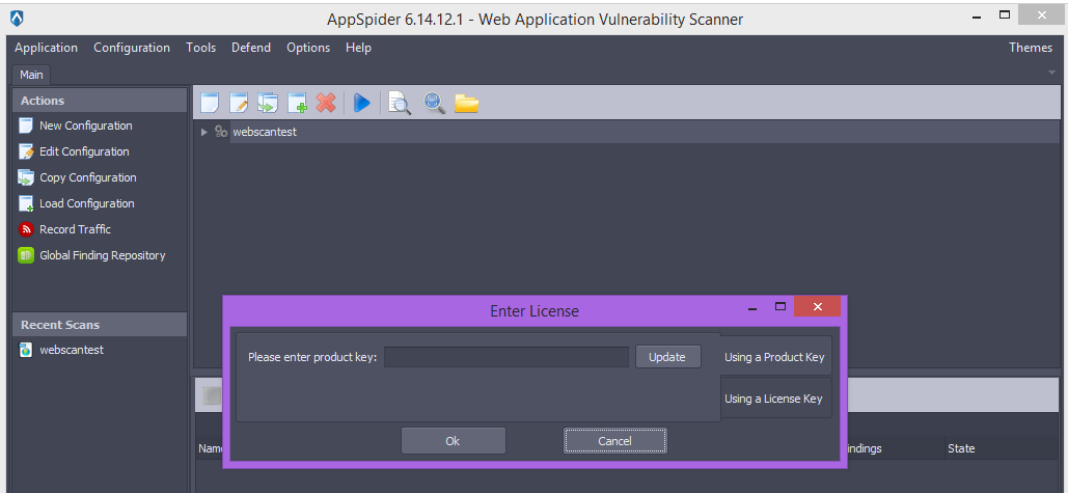
- **Environment** - opens *Environment* tab with the following pages:
  - **Scan data directory** - directory settings for scan data
  - **Traffic Recording** - opens settings for Traffic Recording (*Max size of Traffic log* counter)
  - **Macro Recording** - manages settings of the Macro recorder
  - **Auto Update** - manages settings of application updates (Auto Update options and User credentials)

Help

- **User guide** - opens *AppSpider User Guide*
- **License** - opens *License Information* window

To activate your license:

1. Click the Help>License> **Update License** button.
2. Input the Product Key which was e-mailed to you from Rapid7. If you did not received your Product Key then contact support@rapid7.com



Themes

- List of AppSpider user interface themes. You can select a different theme.

Actions Panel

The Actions panel on the main window contains the following actions:

- **New Configuration** - creates a new configuration
- **Edit Configuration** - edit the selected configuration
- **Copy Configuration** - copies the selected configuration to the List of scan configurations panel
- **Load Configuration** - opens a configuration from the file system
- **Record Traffic** - records traffic

List of scan configurations

This panel contains the list of all scan configurations. You may expand a scan configuration to observe the scan reports. Double-clicking on a report will open the HTML version with complete details about the scan and vulnerabilities found.

Scan configuration actions

The following buttons are available in this toolbar:

- **New** - opens a tab for creating a new scan configuration
- **Edit** - opens a tab for editing the selected configuration
- **Copy** - copies the selected configuration
- **Load** - loads a scan configuration from the file system
- **Remove** - removes the selected scan configuration
- **Run** - starts a new scan based on the selected configuration
- **View Report** - view the last HTML report for the selected scan or for the selected scan result
- **View Scan Results** - view the last report in the Scan Console for the selected scan or for the selected scan result
- **Open folder in Explorer** - opens configuration folder in explorer

Recent scans

The *Recent scans* panel displays the list of recently completed scans. Highlights the scan configuration on the List of scan configurations panel when clicked. Double-click opens HTML report in browser.

Active scans

The *Active scans* panel displays the list of active scans. Double-click opens the status of the clicked scan.

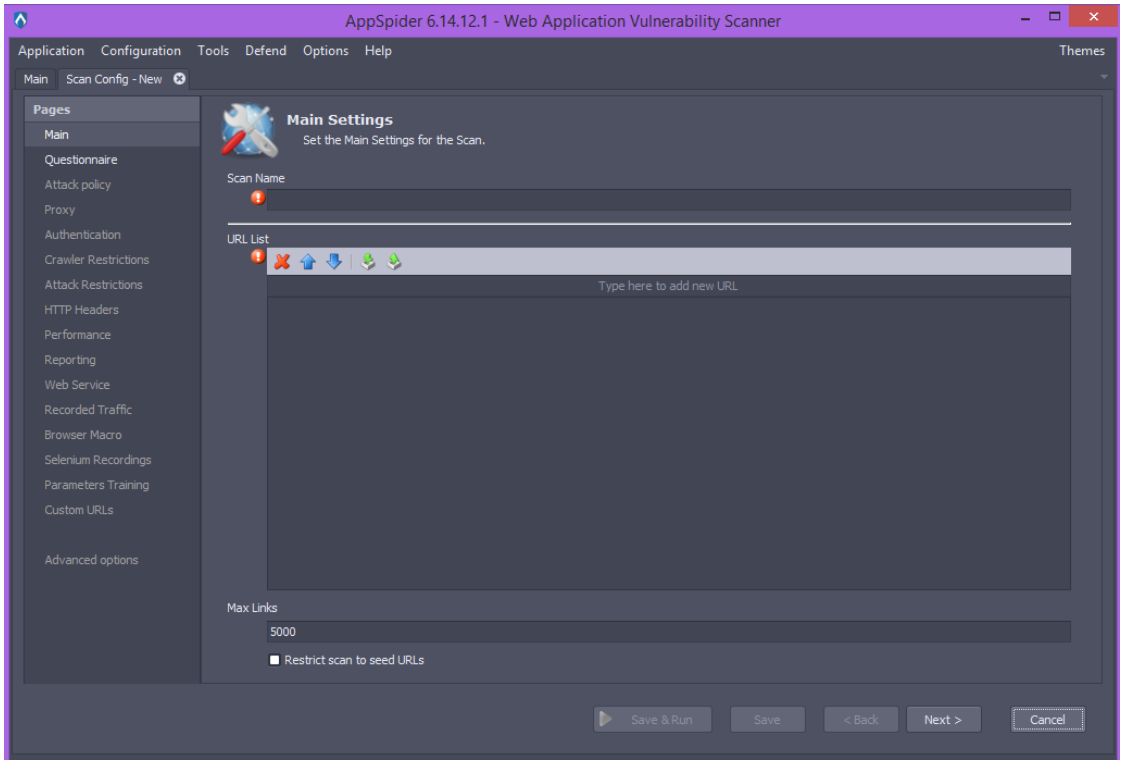
Active scan actions

The following buttons are available in this toolbar:

- **Pause**- pauses the selected scan
- **Stop** - stops the selected scan
- **Status** - opens a tab for the selected scan

New Configuration

The *Scan Config- New* window allows you to create a new scan configuration.

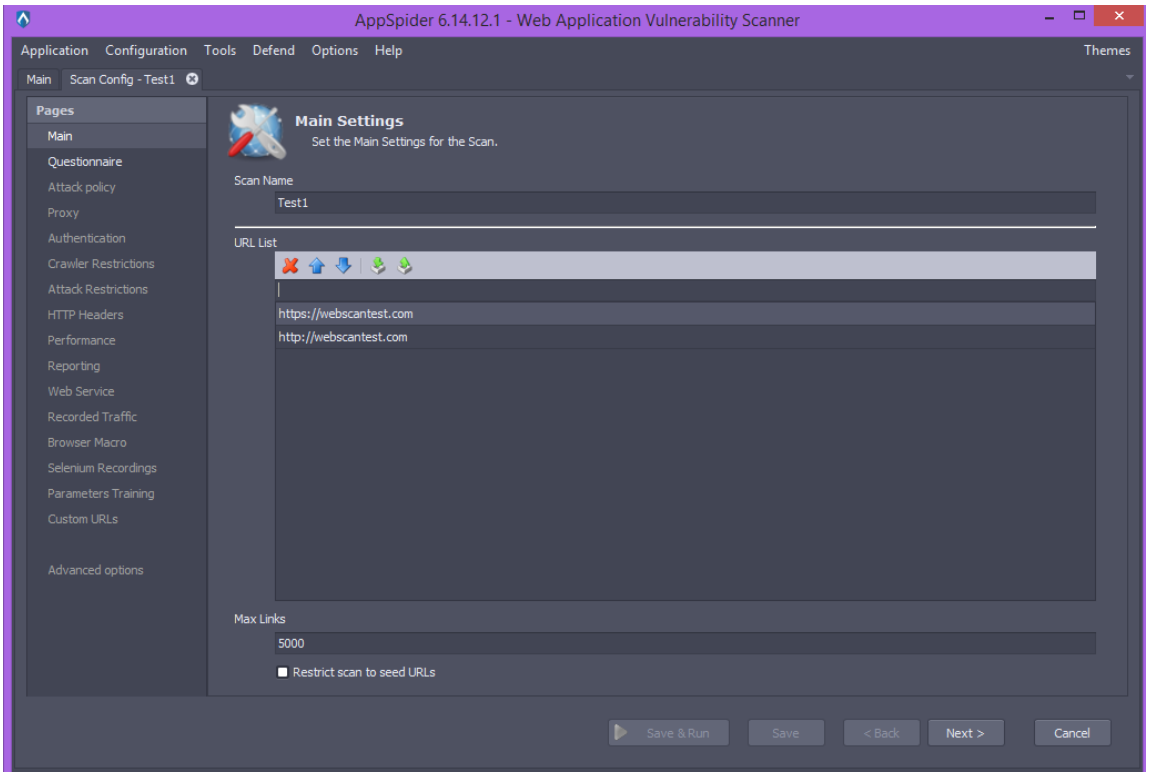


The following panels are presented on the New Configuration setup screen:

- *Pages*: the list of all possible scan configuration settings. The *Main Settings* are selected by default. You can't switch to another page until a Scan Name is chosen and at least one URL entered on the *Main Settings* page.
- **Save & Run**: saves the scan configuration and starts the scan for this configuration.
- **Save** button: saves the scan configuration.
- **Back** button: returns to the previous panel.
- **Next** button: goes to the next panel.
- **Cancel** button: goes to main screen without saving.

Main Settings

This panel allows you to enter a scan name and a list of URLs to scan.



The following elements are available on this panel:

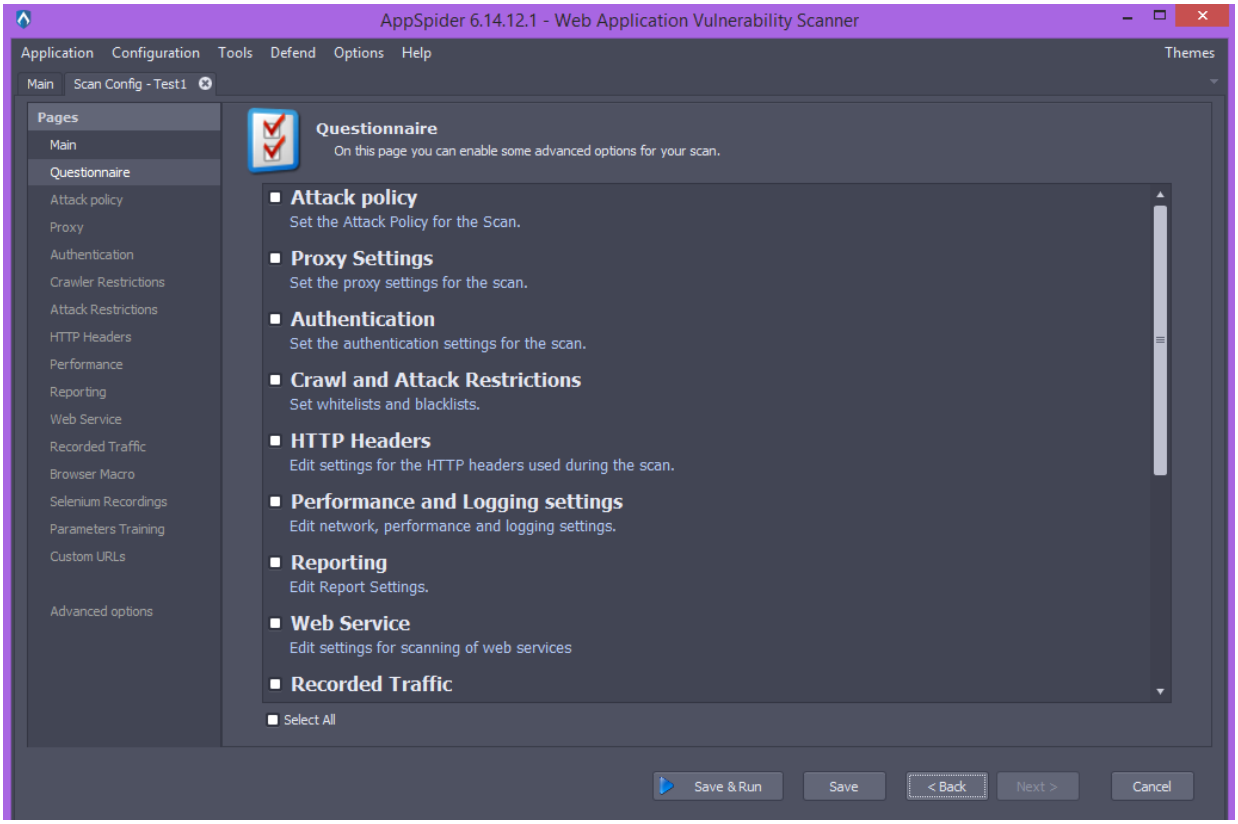
- **Scan Name**: the name of the scan configuration.
- **URL List**: panel to enter a list of the scanned URLs. On entering a URL without a schema (http:// or https://) two URLs are added the first with http and the second with https schemas.
  - You may enter URLs one by one using the text field.
  - You may import URLs from file system using the **Import** button. The file should have a .txt extension and contain one URL per line.
  - You may export URLs to the file system using the **Export** button.
  - You may arrange URLs by selecting any URL in the list and pressing the **Move Up**/**Move Down** buttons. URLs will be crawled in the order displayed and this may impact scan performance if the application is expecting links to be executed in a particular order.
  - You may remove any URL from the list by selecting the URL and pressing the **Delete** button.
- **Max Links**: maximum number of links to crawl.
- **Restrict scan to the seed URLs** checkbox: This will limit the links crawled to those listed. No additional links will be crawled.

Questionnaire

This panel allows you to enable advanced options for the scan configuration. The list of advanced options is displayed.

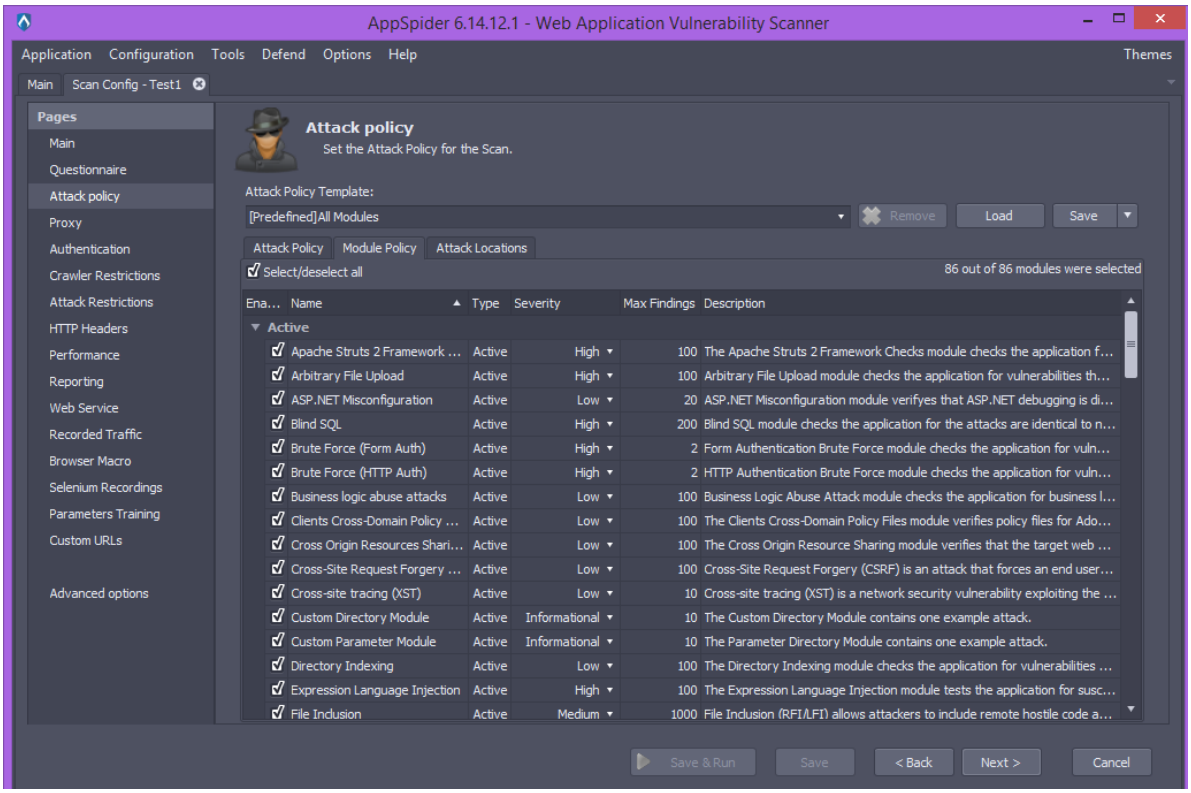
The *Main* and *Questionnaire* pages are enabled by default. A **Select all** option is available.

If there are no additional configurations such as authentication that are needed then the "Save & Run" button can be selected to kick off the AppSpider Scan.



Attack policy

This panel allows you to select the list of attack types, attacks locations (such as directory, file, or path) and other properties.



The following elements are presented on this panel:

Attack Policy Template

This contains the types of predefined attack policies and a **Load** button to load the selected template:

- **All Modules**: selects all modules.
- **Crawl Only**: deselects all modules.
- **Passive analysis**: selects modules for passive analysis.
- **SQL Injection**: selects SQL Injection modules
- **XSS**: selects cross-site scripting modules
- **SQL Injection and XSS**: selects SQL Injection and cross-site scripting modules

You can save and load the custom/user attack policies with the **Load** and **Save/Save as** action buttons.

The user policies are located in the AppSpider folder under *AttackPolicies* folder. Each policy is located in a separate file.

If the policy name matches a scanner default policy name, then the user policy overwrites the scanner default policy.

The interface merges default scanner policies with user policies in the Attack policy template list.

The **Save** button allows you to save currently selected modules as a new policy.

Attack Policy tab

- **Attack Policy Name**: editable field for Policy name
- Attack Prioritization
  - **Sequential**: run all attacks on the first link, then switch to the next link.
  - **Randomized**: randomly select a link from the attack space.
  - **Smart**: Rapid7 proprietary attack prioritization algorithm that prioritizes attacks.
- Attacks per input:
  - **All**: all potential attacks are made.
  - **Smart**: Rapid7 uses a proprietary attack selection to minimize redundant attacks.
- Attacks Collection:
  - **All**: all attacks from all collections.
  - **Smart**: Rapid7 uses a proprietary algorithm to select the best attack collection to run for a particular attack point.
- Browser Encoding:
  - **No**: Do not do browser encoded attacks.
  - **Yes**: Use the same attack encoding as the browser.
- **False Positive Regex**: define a regular expression to detect a server response that should be treated as a False Positive findings.

Module Policy tab

This contains the list of all scan modules and options to select or deselect modules. Modules are divided into *Active* and *Passive* attacks. The *Module Policy* table contains the following columns:

- **Enabled** checkbox: enables the selected attack module.
- **Name**: name of the attack module.
- **Type**: type of attack module(Active or Passive).
- **Priority**: priority of the attack module. You may change the value of this property.
- **Max Findings**: number of maximum findings for the attack module. You may change the value of this property.
- **Description**: description of the attack module.

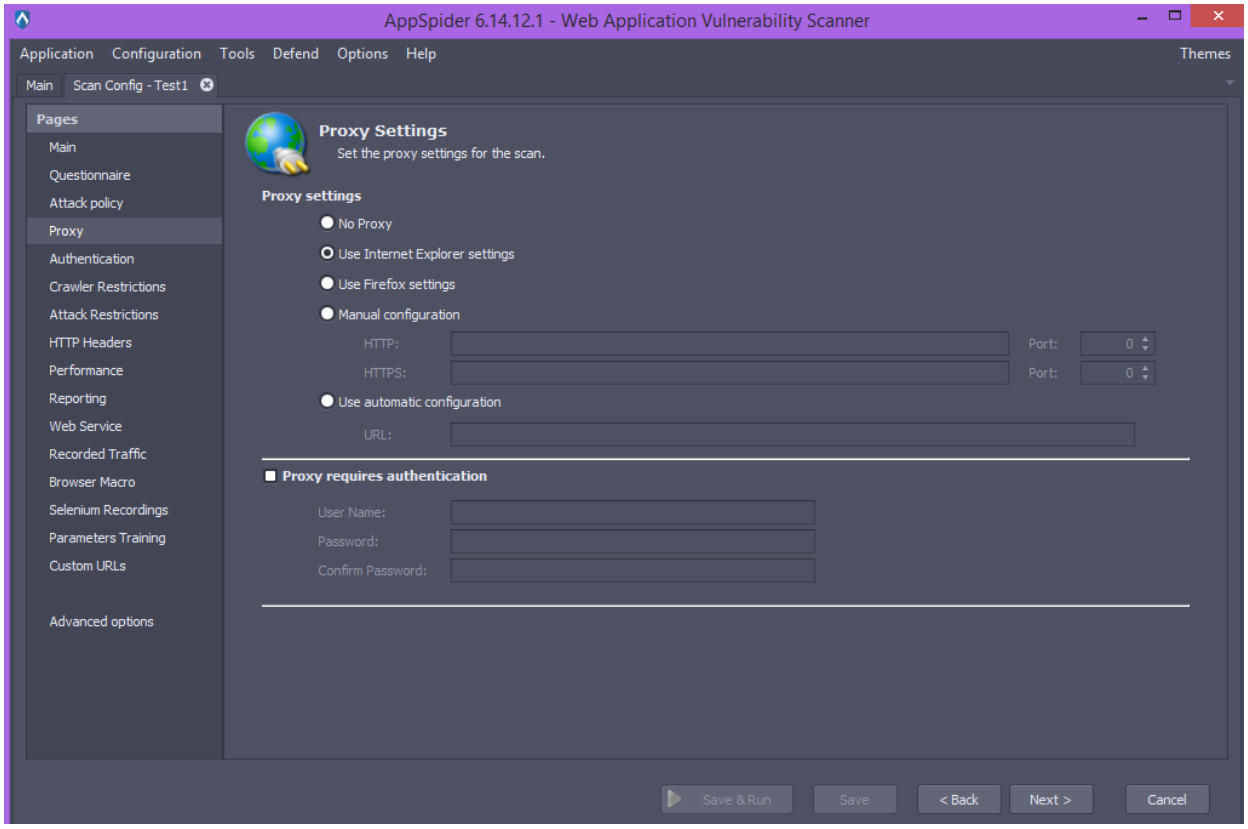
Attack locations tab

This table contains the list of attacks with options to select the attack location (such as directory, path, or query). The table contains the following columns:

- **Attack**: the attack name.
- **Directory**: the checkbox to enable the attack location.
- **File**: the checkbox to enable the attack location.
- **Path**: the checkbox to enable the attack location.
- **Query**: the checkbox to enable the attack location.
- **Fragment**: the checkbox to enable the attack location.
- **Post**: the checkbox to enable the attack location.
- **HTTP Header**: the checkbox to enable the attack location.
- **Cookie**: the checkbox to enable the attack location.
- **Referrer**: the checkbox to enable the attack location.

Proxy

The panel allows to you to set the proxy settings for the scan.



The following options are available:

Proxy settings

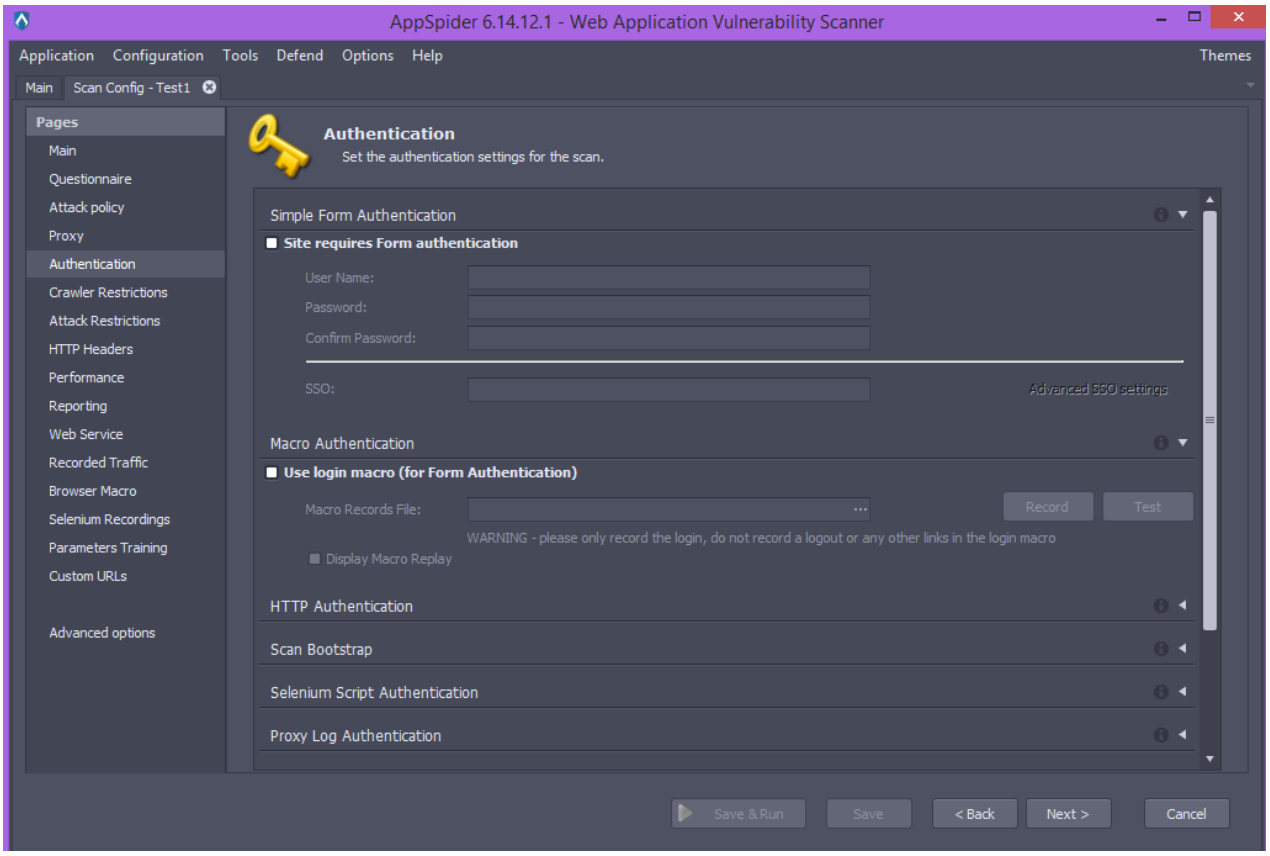
- **No proxy**: disable proxy settings
- **Use Internet Explorer settings**: enable Iinternet Explorer settings for the scan configuration
- **Use FireFox settings**: enable Firefox settings for the scan configuration
- **Manual configuration**: you can manually configure HTTP and HTTPS settings.
- **Use automatic configuration**: you should input a link to the automatic configuration script.

Proxy requires authentication checkbox:

You can mark this checkbox and enter a username and password for the proxy.

Authentication

The panel allows you to set the authentication settings for the scan.



Form Authentication and *Macro Authentication* methods are expanded by default.

Scan Bootstrap, *Redirect to SSO*, *Session Hijacking*, *HTTP Authentication*, *Advanced settings* settings are disabled by default.

The *Form Authentication* method contains the following settings:

- **Site requires Form authentication** checkbox: enables form authentication; you should type in a username and password.
- **SSO**: This is required if a single-signon solution has an login page outside of the domain restrictions set up in the scan.

The *Macro Authentication* method contains the following settings:

- **Use login macro**: enables macro authentication and records or uploads a previously recorded macro from the file system.

The *HTTP Authentication* method contains the following settings:

- **Site requires HTTP authentication** checkbox: you may enable HTTP authentication and enter a username and password to use the previously entered Form credentials.

The *Scan Bootstrap* setting contains **Allow scan bootstrap** checkbox and **Delay (ms)** counter (60000 is predefined).

The **Scan bootstrapping** option allows you to manually log the scanner into the web site, and/or to guide the scanner to a specific area of the website. Scan bootstrapping is done within an embedded Internet Explorer browser. Delay setting sets the time limit that the browser is open for you to interact with the website.

The **Redirect to SSO** setting contains the **Allow initial redirect to SSO** (no forms used) checkbox.

Session Hijacking setting contains **Session Hijacking** checkbox and **Utilize Captured Session Cookie** link. The link opens *Session* settings with the following fields:

- **Session Cookie** text field
- **Lock cookie** values for duration of scan checkbox
- **Apply to all cookies** checkbox.

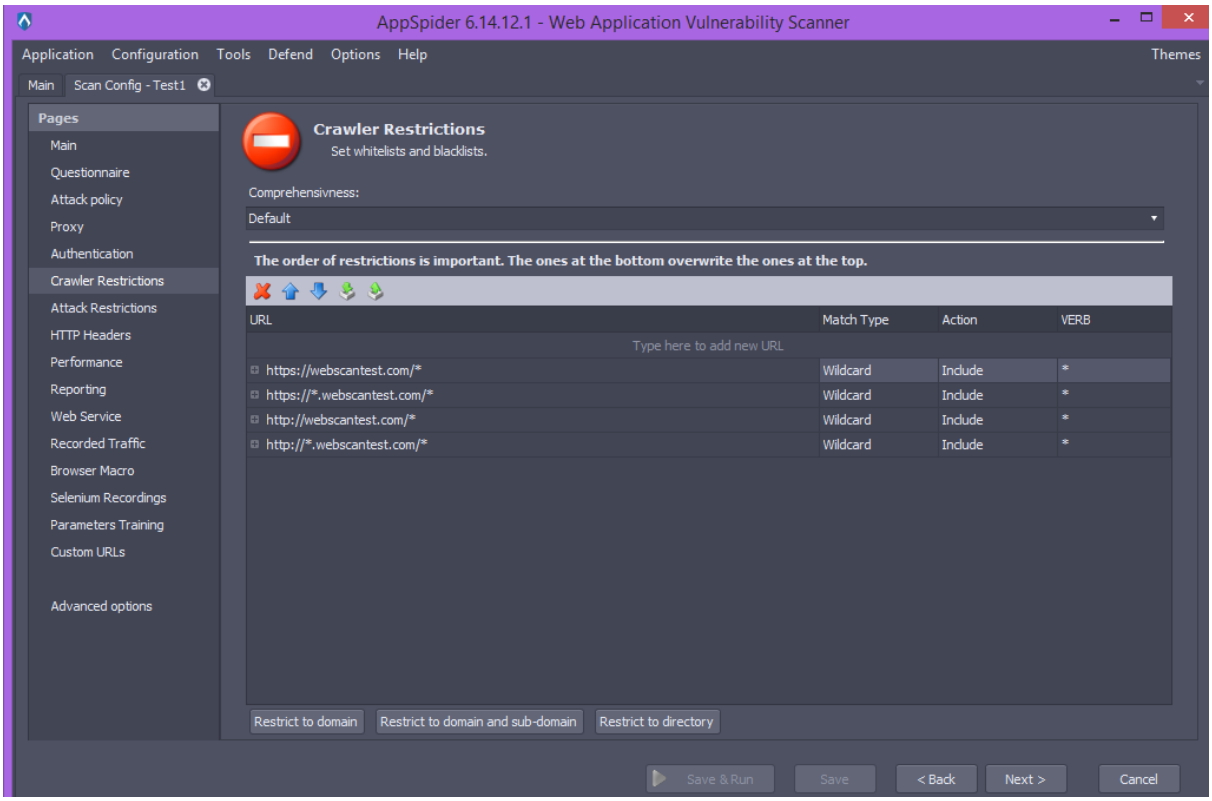
Action buttons are **Ok** and **Cancel**.

Advanced settings contains:

- **Configure SSL certificate** link: opens *SSL configuration* window where user may configure the SSL certificate
- **Logged IN Regex** text field
- **Assume Good login** checkbox. This eliminates AppSpider checking to see if the login appears to work. Occasionally, these checks can invalidate the user account.

Crawler Restrictions

This panel allows you to set a whitelist and blacklist for crawling by entering URLs into the URL field. You may remove and edit the order restrictions using the **Delete**, **Move Up** and **Move Down** buttons.



Comprehensiveness

If the Fast Scan option is selected AppSpider will attempt to not crawl/attack duplicative URLs in an attempt to improve scan speeds. In the Default mode all URL's will be crawled/attacked.

The table contains the following columns:

URL

URL, URL with wildcard, literal or regular expression.

Match type

This combo box provides the following properties:

- **Wildcard**: use the URL with the following wildcards. * matches any symbol, ?: matches one symbol(e.g. http://www.webscantest.com/*)
- **Literal**: will only match the exact string
- **Regex**: use a regular expression

Action

This combo box provides the following properties:

- **Include**: include the URL in the crawl.
- **Exclude**: exclude the URL from the crawl and do not scan it.

The following action buttons are presented:

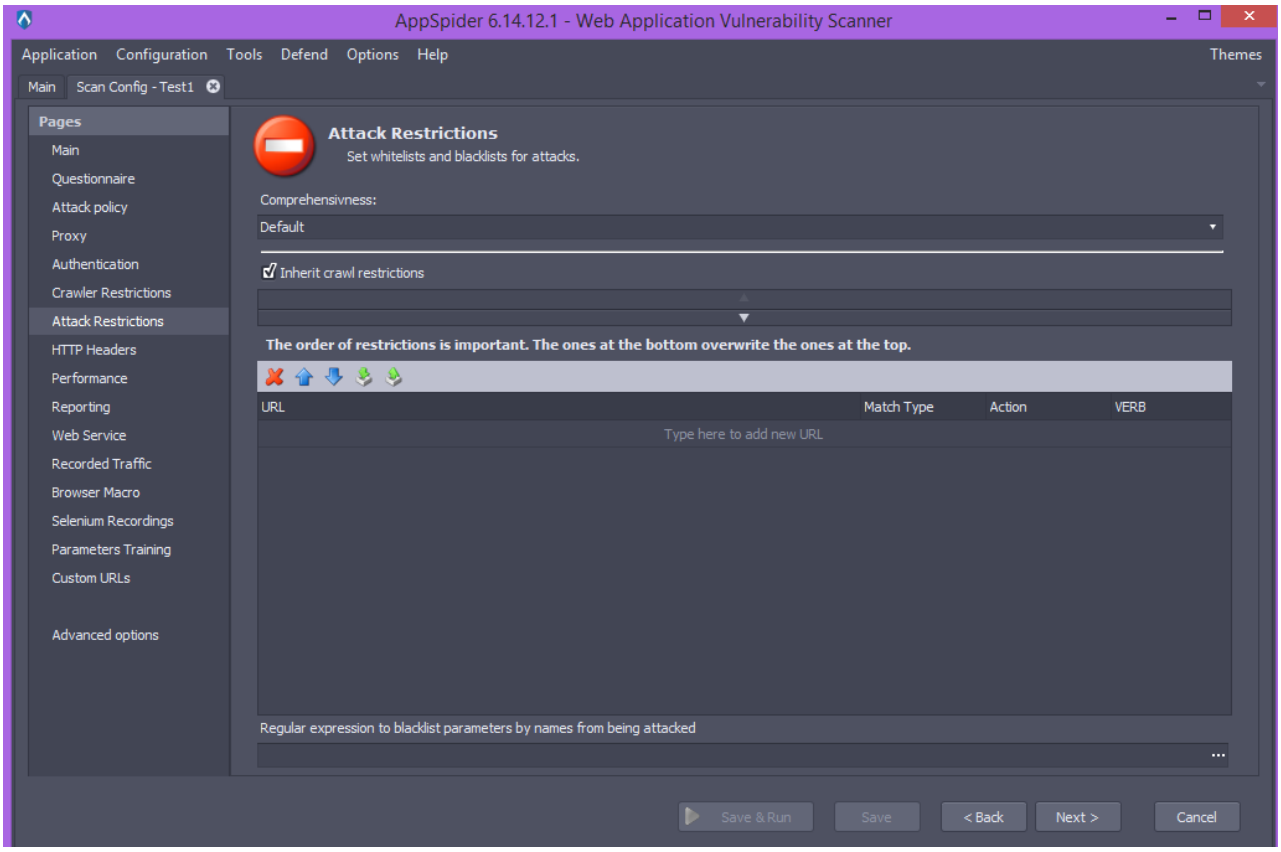
- **Restrict to domain**: reset the restriction list and update it with restrict to the domain properties. (http://www.webscantest.com/datastore/:> http://www.webscantest.com/*)
- **Restrict to domain and subdomain**: reset the restriction list and update it with the restrict to domain and subdomain properties. (http://www.webscantest.com/datastore/:> http://*.www.webscantest.com/*)
- **Restrict to directory**: reset the restriction list and update it with restrict to the directory properties. (http://www.webscantest.com/datastore/:> http://www.webscantest.com/datastore/*)

Verb

Verb is a Web Verb (sometimes they called methods): GET, POST, HEAD, DELETE, etc. By default, the restrictions are applied to all verbs, but we have decided to allow user the flexibility to provide restrictions specific to a particular verb. So for example, user can disable a DELETE completely.

Attack Restrictions

The panel allows you to set the whitelist and blacklist for attacks by entering a URL into the table. You may remove and order the restrictions using the **Delete**, **Move Up**, and **Move Down** buttons.



The table contains the following columns:

- **URL**: URL, URL with wildcards, literal, or regular expression.
- **Match type**: combo box with the following properties:
  - **Wildcard**: use a URL with the following wildcards: * matches all symbols and ?matches one symbol (e.g., http://www.webscantest.com/*)
  - **Literal**: will only match the exact string listed
  - **Regex**: use a regular expression
- **Action**: combo box with the following properties:
  - **Include**: Include the URL into the scan and run attacks against it.
  - **Exclude**: Exclude the URL from attacks
- **Verb:** Verb is a Web Verb (sometimes they called methods): GET, POST, HEAD, DELETE, etc. By default, the restrictions are applied to all verbs, but we have decided to allow user the flexibility to provide restrictions specific to a particular verb. So for example, user can disable a DELETE completely.
- **Regular expression**: Used to blacklist parameters by names from being attacked. You may enter a regular expression into the text field or open the Regex builder by clicking the button.

HTTP Header

The panel allows you to edit the settings for the HTTP headers used during the scan.

The following properties are available:

HTTP header settings

- **Protocol**: combo box with HTTP/1.1 and HTTP/1.0 options.
- **User-Agent**: combo box with Chrome: Mozilla 5.0..., Firefox: Mozilla 5.0... and IEv9: Mozilla 5.0... options.
- **Accept Headers**: text box with the ability to select some values using a drop down list.
- **Accept Charset**: text box with the ability to select some values using a drop down list.
- **Accept Language**: text box with the ability to select some values using a drop down list.
- **Extra Headers**: text area. Headers should be entered in the following format: aaa: bbb. Multiple header are line break separated.
- **Accept Encoding**: text box with ability to select some values using a drop down list.
- **Cookie**: text area.
- **Use session Hijacking to login to website** checkbox: when checked, cookies from the Cookie text area will be used for session hijacking.

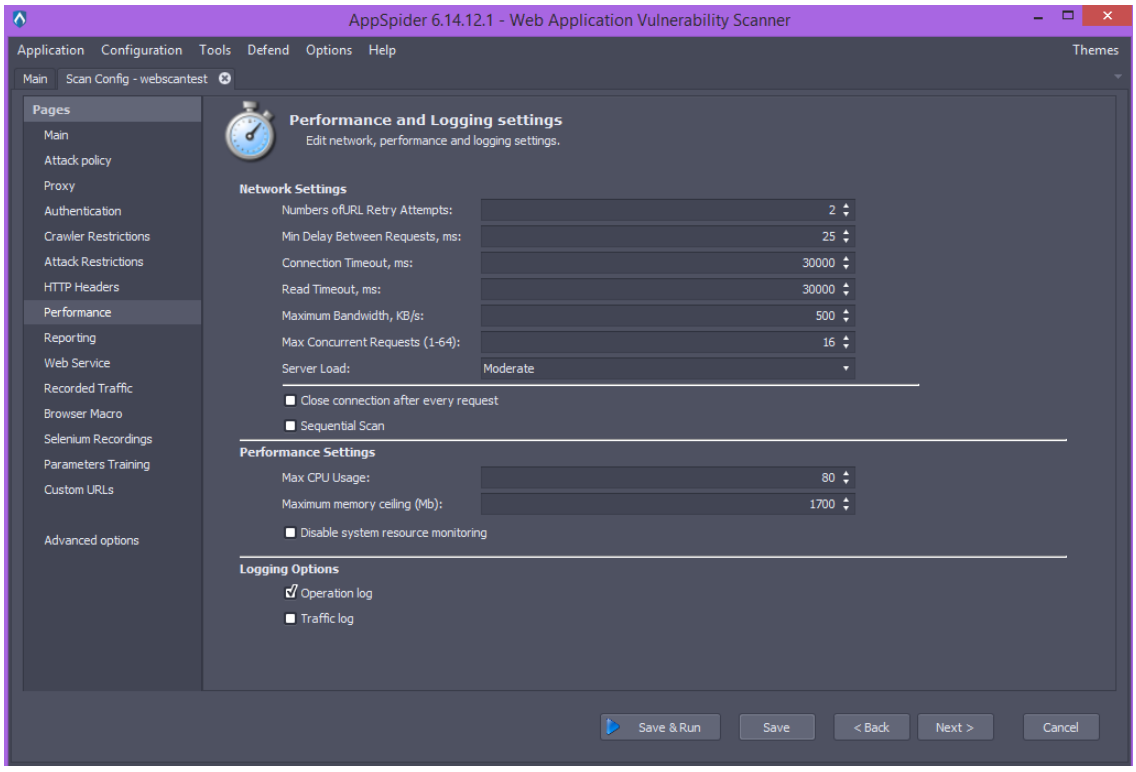
Lock cookie values for duration of scan checkbox

When marked the following properties become active:

- **Apply to cookies** checkbox: when checked, the following properties become active:
- **Cookies to lock**: text area with ability to add, delete, sort, export and import cookies. Cookie format is;
    - HTTP Headers format:
    - a=6; b=7

Performance

The panel allows you to edit the network, performance and logging settings.



The following elements are available on the panel:

Network settings:

- **Number of URL Retry Attempts**: text field
- **Min Delay Between Requests, ms**: text field
- **Connection Timeout, ms**: text field
- **Read Timeout, ms**: text field
- **Maximum Bandwidth, KB/s**: text field
- **Max Concurrent Requests (1-64)**: text field
- **Server Load**: is combo box with *light*, *moderate*, *heavy*, and *custom* preset values for minimum delay between requests, maximum bandwidth and maximum concurrent requests. These are displayed when any of these options are selected. *Moderate* is selected automatically when you change any of the options.
- **Close the connection after every request** checkbox
- **Sequential Scan**: run all the attacks on the first link before switching to the next link

Performance settings:

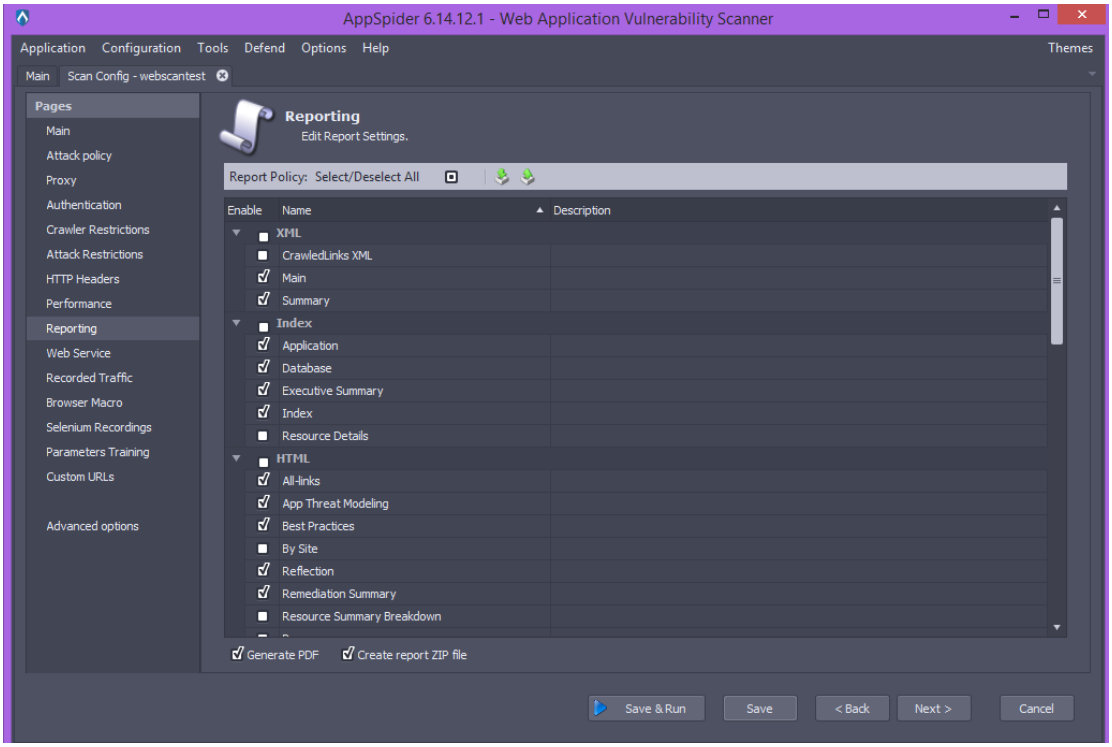
- **Max CPU Usage (%)**: text field
- **Maximum memory ceiling (Mb)**: this is the maximum memory that AppSpider will allow itself to take up before it shuts itself down.
- **Disable available memory monitoring**: this stops AppSpider from checking if it has hit the maximum memory ceiling. If AppSpider does run out of memory, this could cause a crash.

Logging options

- **Operation log**: the operation log details the actions taken by AppSpider (e.g. crawling a link, making a specific attack, etc.).
- **Traffic log**: the traffic log details the request response traffic. This is very helpful if debugging is required. These logs can become very large with longer scans.

Reporting

This panel allows you to configure the reports produced from a scan.



The following settings are available:

XML

You may select the following options to add to the report:

- **Main**
- **Summary**

Privacy

- **Comments**
- **Cookies**

HTML

You may select the following options to add to the report:

- All-links
- App Threat Modeling
- Best Practices
- By Site
- Reflection
- Remediation Summary
- Resource Summary Breakdown
- Server
- Site Links
- Status And Config
- Vulnerabilities
- Vulnerabilities By Url Standalone

Compliance

You may select the following options to add to the report:

- CWESANS
- DISASTIG
- FISMA
- GLB
- HIPAA
- OWASP2007
- OWASP2010
- OWASP2013
- PCI
- PCI31
- SOX

Advanced

- **Crawled Links XML**
- **IncludeDbinZip**

Generate PDF checkbox

Create a PDF of the report.

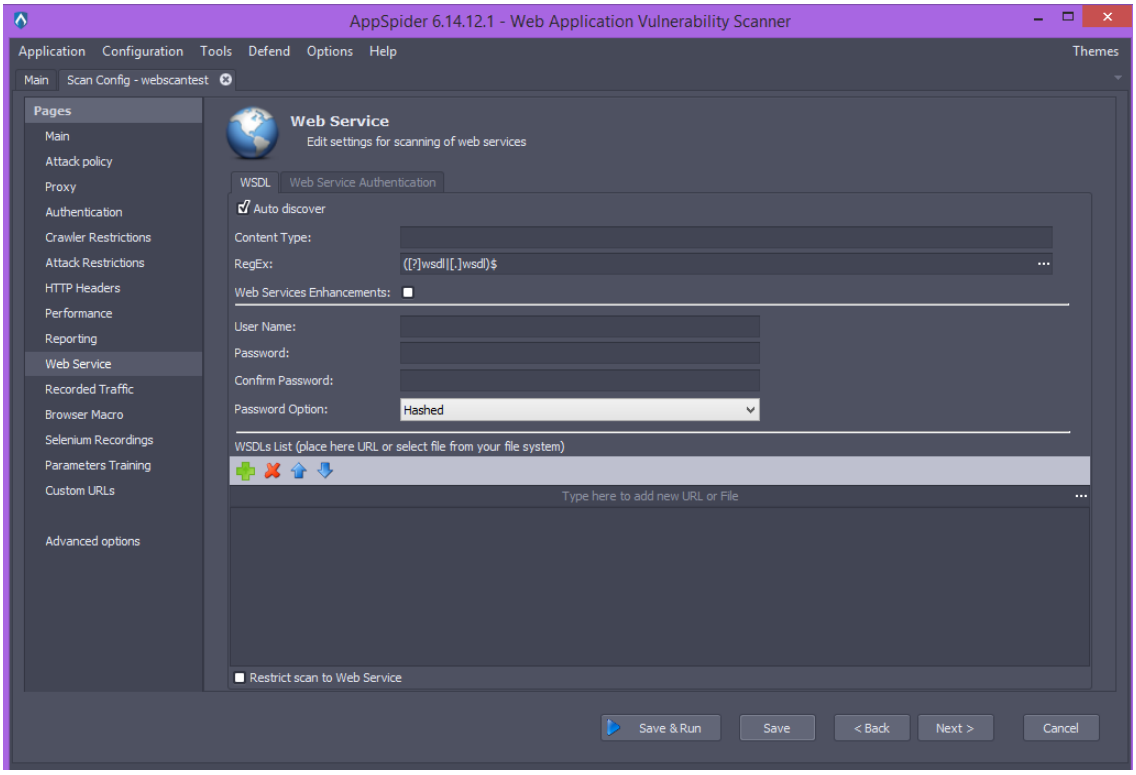
Create report ZIP file checkbox

Create a ZIP archive of the report.

Report output files will be stored in the scan report directory which is typically located at Documents>AppSpider>Scans>Scan Config Name>Scan Date.

Web service

The panel allows you to edit the settings for scanning of web services.



WSDL tab

The *WSDL* tab contains the following elements:

- **Auto discover** checkbox: enables the service auto discover functionality.
- **Content type** text field: used to send to the web server in the SOAP requests when the scanner fails to extract the content-type information from the WSDL.
- **Regex** text field: regular expression to identify URLs that host WSDL files.
- **Authentication settings**: *User name*, *Password*, and *Confirm password*.
- *WSDLs List*: you may add the URLs of WSDL services. You can use the browse button to import files located on the web or the local file system.
- **Restrict scan to Web service** checkbox
- Action buttons for *Add*, *Delete*, and *Sort*.

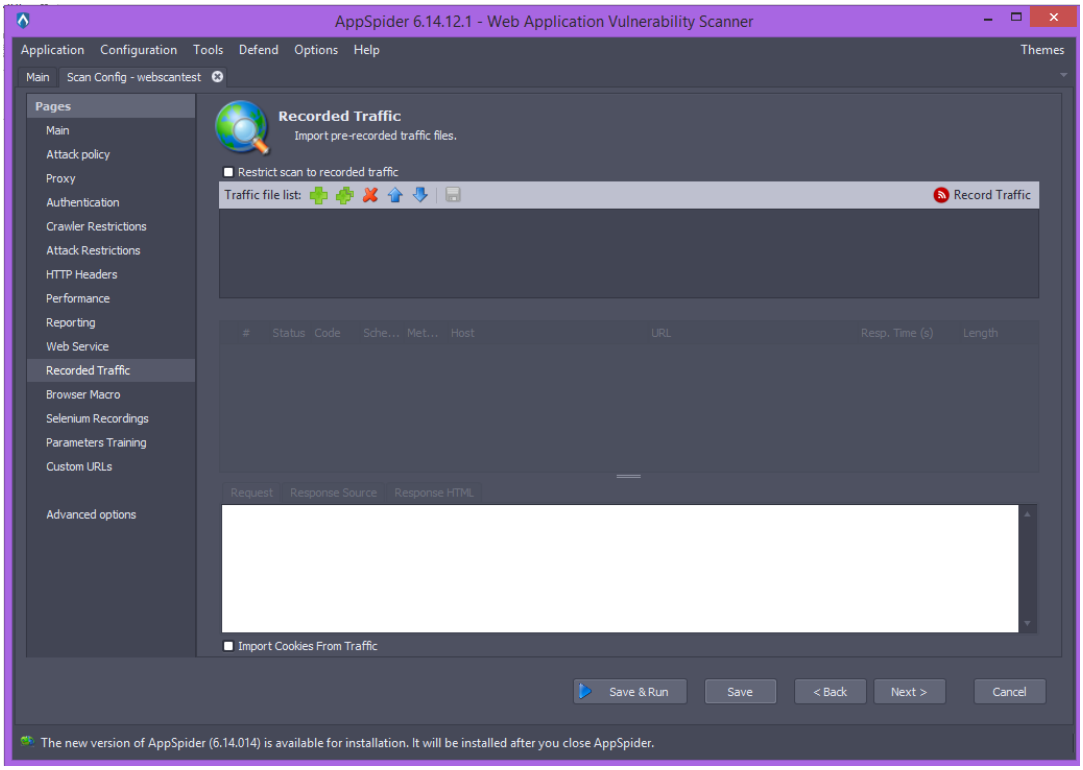
Web Service Authentication tab

The *Web Service Authentication* tab contains the following elements:

- *Custom web service authentication* checkbox: enables the web service authentication
- *Web service* selection list
- *Authentication web method* selection list
- *<Authenticate method name> Request* table: *Parameter*, *Value*, *Encrypt* columns.
- *Extract and apply authenticate token* checkbox
- *Extract* button
- *Auth Token parameter name in Authenticate method response* selection list
- *Auth Token parameter name in regular (non-auth) method response* selection list

Recorded Traffic

This panel allows you to record or import pre-recorded traffic files.



The panel contains the following items:

- *Restrict scan to recorded traffic* checkbox: If this option is selected, scanning is performed only on URLs from the imported traffic file.
- *Traffic file list*: the list of imported traffic files.

  The following options are available in the toolbar:

  - **Add**: add a traffic file from the file system.
  - **Bulk Import**: add all traffic files from the selected file system directory. Opens the *Bulk files import* window.
  - **Delete**: removes a selected traffic file from the list.
  - **Sort**: sorts the traffic files in the list.
  - **Save**: saves changes in selected traffic file.
  - **Record Traffic**: button opens the *Traffic Recorder* tab. If you close the tab after recording and saving the traffic, a file automatically gets added to the *Traffic file* list.

  The table contains the following columns:

- *Index*: the number of the log entries
- *Status*: the status of the request
-  : Code = 0. Request failed. The server did not send any response
-  : Request succeeded. The response from the server was received.
- *Code* - response code
- *Host* - the host
- *Url* - the host's relative URL
- *Resp. time*: the response time
- *Length*: the response length

Selecting any row will show the user detailed information in the tabs under the table.

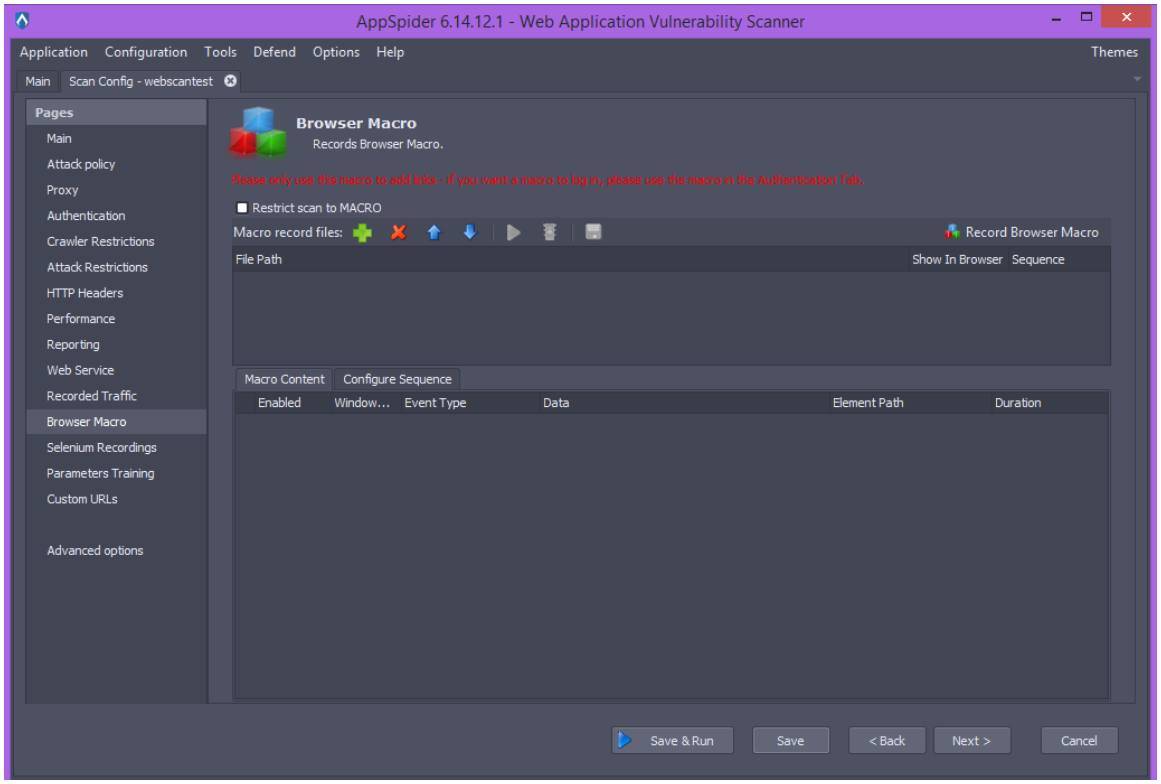
- *Request* - the request that was sent
- *Response Source* - the response source
- *Response HTML* - the response html

- **Import cookies from traffic** checkbox presented in the bottom of tab.

Browser Macro

The panel allows the user to record or import pre-recorded macro files.

A macro is a sequence of actions (e.g. menu selections, link executions, value entries, etc.) that will be replayed exactly as input by the user.



The panel contains the following elements:

- *Restrict scan to macro* checkbox: this will result in AppSpider only crawling and testing the pages/actions executed in the Macro. No other pages will be crawled or tested.

Macro record files: the list Macros added for the users macro file. The list contains the following options:

- *Add* - add a macro file from file system.
- *Delete* - removes a selected traffic from the list.
- *Sort* - sorts the traffic files in the list.
- *Test macro* - test if macro executed successfully or not.
- *Configure sequence* - opens “Configure sequence” tab.
- *Save changes* - saves changes in selected traffic file.
- *Record Macro* button - opens the *Browser Macro Recorder* tab.

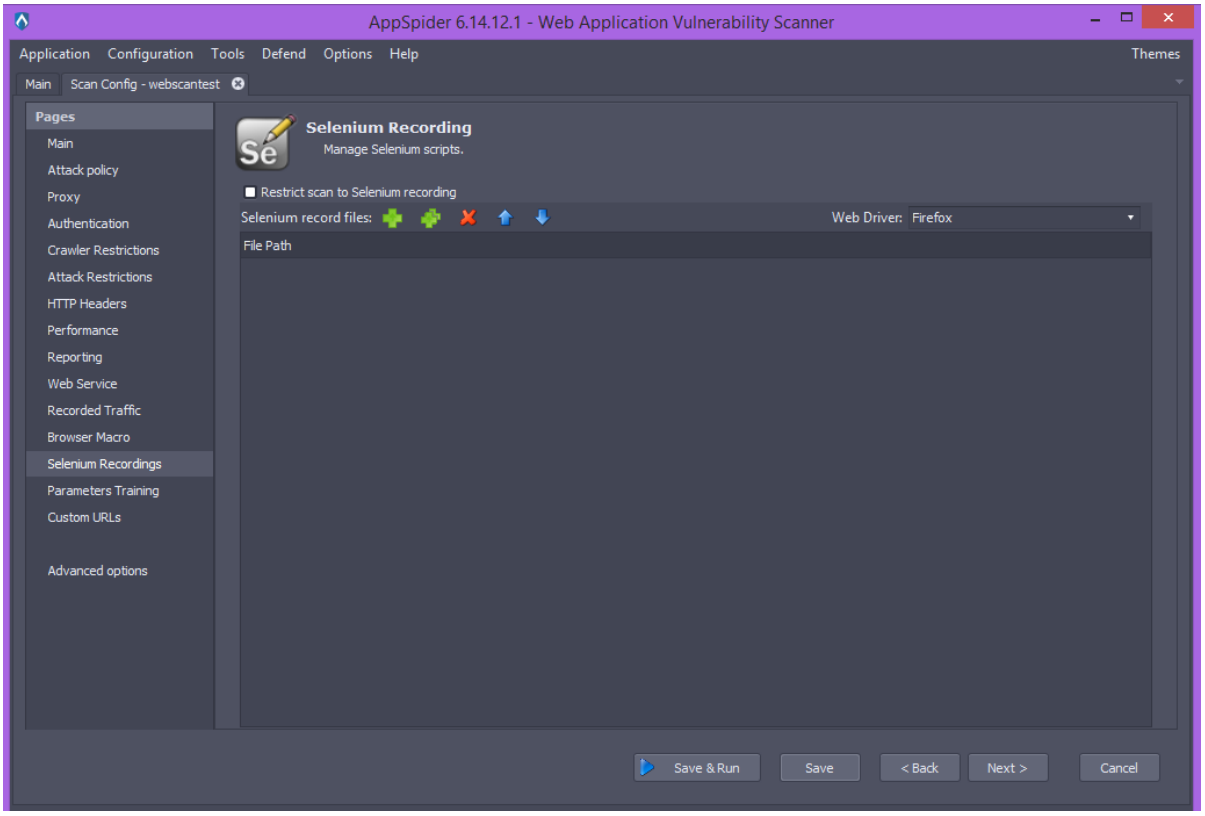
- The following Macro file parameters are presented in the grid:
  - *File path*: the macro's directory path
  - *Show in browser* checkbox - shows the macro record in the browser window when the macro file is played
  - **Sequence** checkbox - enables the *Configure sequence* tab.
- The *Macro content* tab contains a grid with the following macros parameters: *Enabled*, *Window index*, *Event type*, *Data*, *Element path*, and *Duration*.
- The *Configure sequence* tab contains a grid with the following parameters: *URL*, *Attack*, *Send*, and *From cache*.

  The table and *Read request* button are enabled if *Auto Select request* is unchecked.

  Use the *Test sequence* button to test if sequence executed successfully or not.

Selenium Recordings

This panel allows you to manage imported Selenium files.

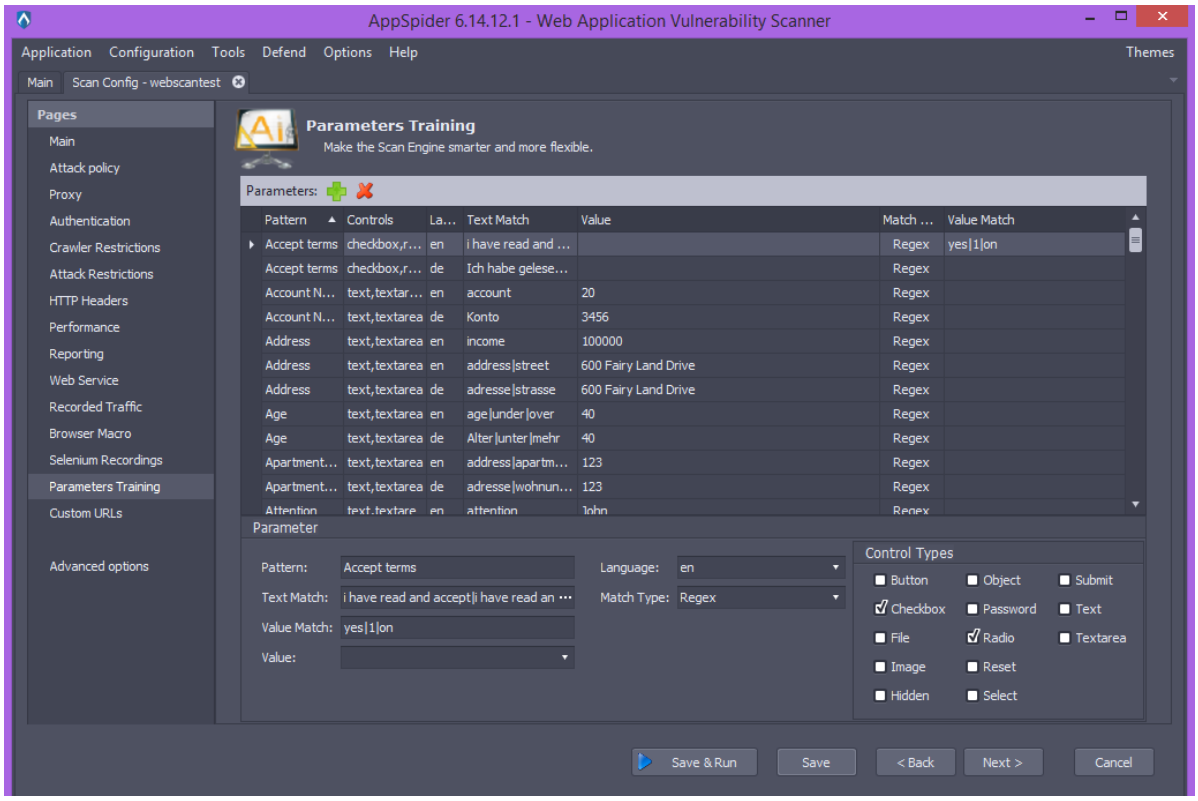


The panel contains the following elements:

- **Restrict scan to Selenium recording** checkbox: selecting this option will result in AppSpider only crawling and testing the pages/actions executed in the Selenium script. No other pages will be crawled or tested.
- **Selenium record list**: the list of imported scripts. The list contains following options:
  - **Add** : add a Selenium file from file system.
  - **Bulk Import** : add all Selenium files from selected file system directory. Opens "Bulk Files Import" window.
  - **Delete** : removes a selected Selenium file from the list.
  - **Sort**: sorts the Selenium files in the list.
- The *File path* parameter presented for the imported Selenium file.

Parameters Training

This panel allows you to manage the scan engine parameters.



The panel contains the following elements:

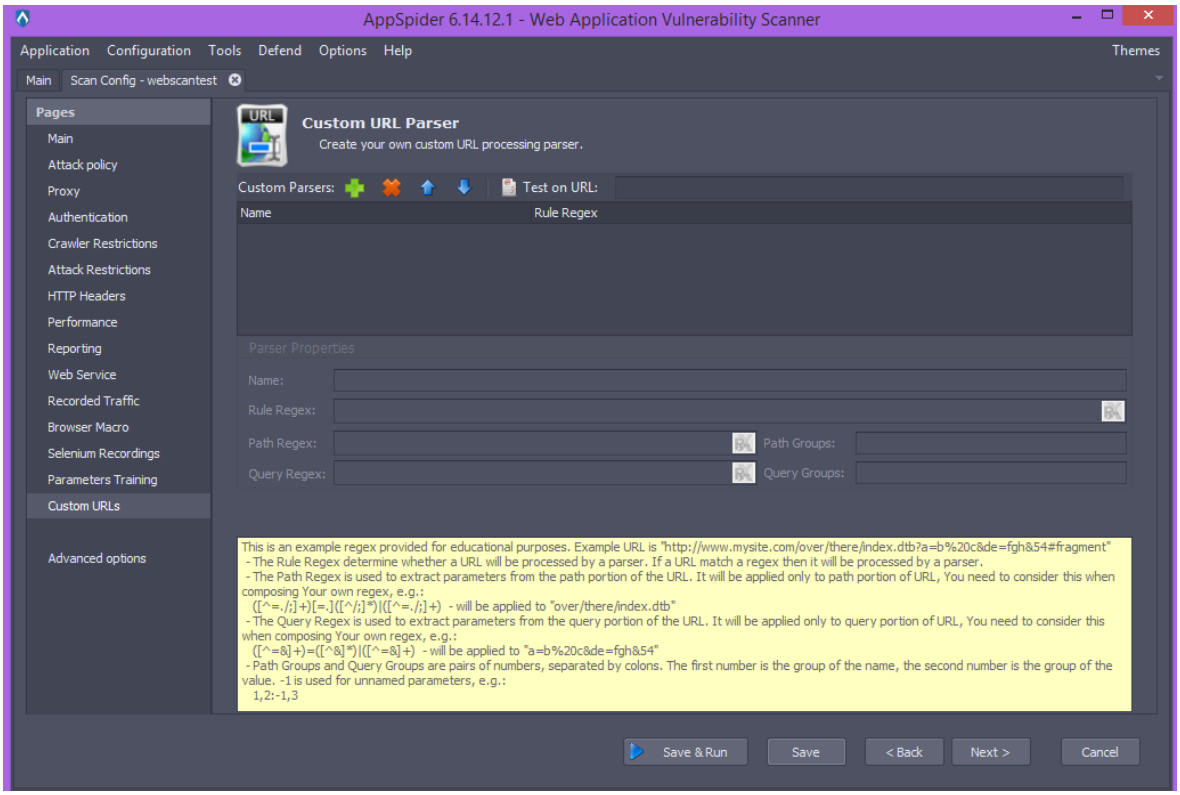
- Action buttons:
  - **Add**: Adds a new parameter to the grid.
  - **Delete**: Deletes the selected parameter from the grid.

- Paramters grid containing the following columns:
  - *Pattern*
  - *Controls*
  - *Language*
  - *Text Match*
  - *Value*
  - *Match type*
  - *Match value*

To modify parameter data, select the parameter and fill in the data in the Parameter area at the bottom of the panel. The fields reflect the columns displayed in the grid. You can also select checkboxes for control types.

Custom URLs

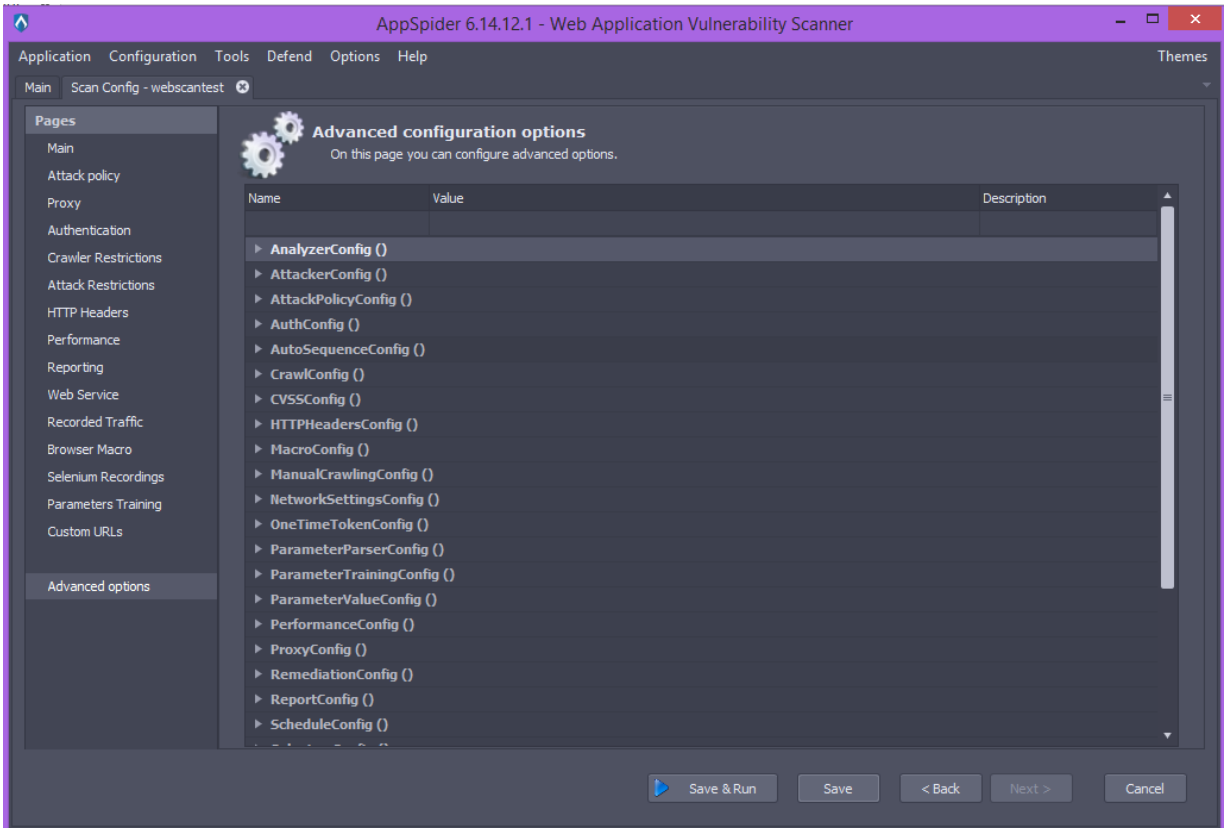
This panel allows you to use the custom URL processing parser.



- *Custom parser* action buttons are:
  - **Add**: add a parser
  - **Delete**: remove the selected parser from the list
  - **Sort**: sort the parsers in the list
  - **Test on URL** text field
- The custom parser table contains the following columns: *Name* and *Rule Regex*.
- Parser properties: *Name*, *Rule Regex*, *Path Regex*, *Query Regex*, *Path Groups*, *Query groups*.

Advanced options

The panel allows you to configure the advanced options. You may sort the settings by name, value, or description.


AppSpider 6.14.12.1 - Web Application Vulnerability Scanner
Application
Configuration
Tools
Defend
Options
Help
Themes
Main
Scan Config - webscantest
Pages
Main
Attack policy
Proxy
Authentication
Crawler Restrictions
Attack Restrictions
HTTP Headers
Performance
Reporting
Web Service
Recorded Traffic
Browser Macro
Selenium Recordings
Parameters Training
Custom URLs
Advanced options
Advanced configuration options
On this page you can configure advanced options.
Name
Value
Description
AnalyzerConfig ()
AttackerConfig ()
AttackPolicyConfig ()
AuthConfig ()
AutoSequenceConfig ()
CrawlConfig ()
CVSSConfig ()
HTTPHeadersConfig ()
MacroConfig ()
ManualCrawlingConfig ()
NetworkSettingsConfig ()
OneTimeTokenConfig ()
ParameterParserConfig ()
ParameterTrainingConfig ()
ParameterValueConfig ()
PerformanceConfig ()
ProxyConfig ()
RemediationConfig ()
ReportConfig ()
ScheduleConfig ()
Save & Run
Save
< Back
Next >
Cancel

Scan Status

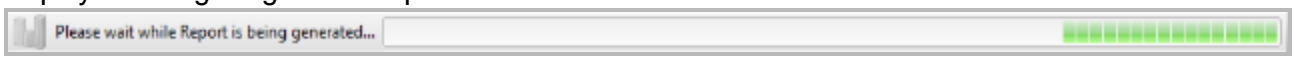
After you start a scan, a Scan Status tab for active scan opens. The default tab is *Status*.

The left panel is collapsible with the  button.

You may stop and pause the scan while it is running() .

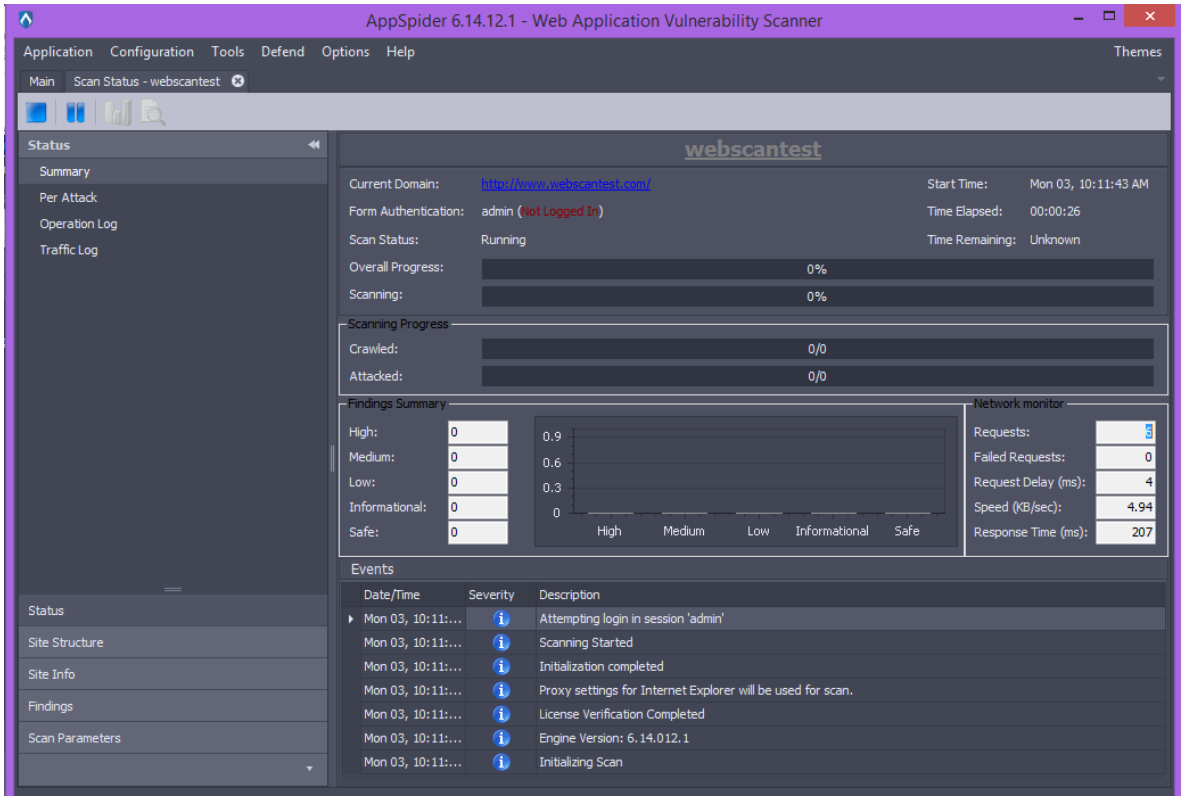
When the scan is in the *Completed* state, you may generate a report ) and "Open HTML Report ()."

The **Generate report** button starts regenerating the report from the database. A "Loading" bar is displayed during the generation process.



Summary

The default selection for the Status page is the scan summary information.



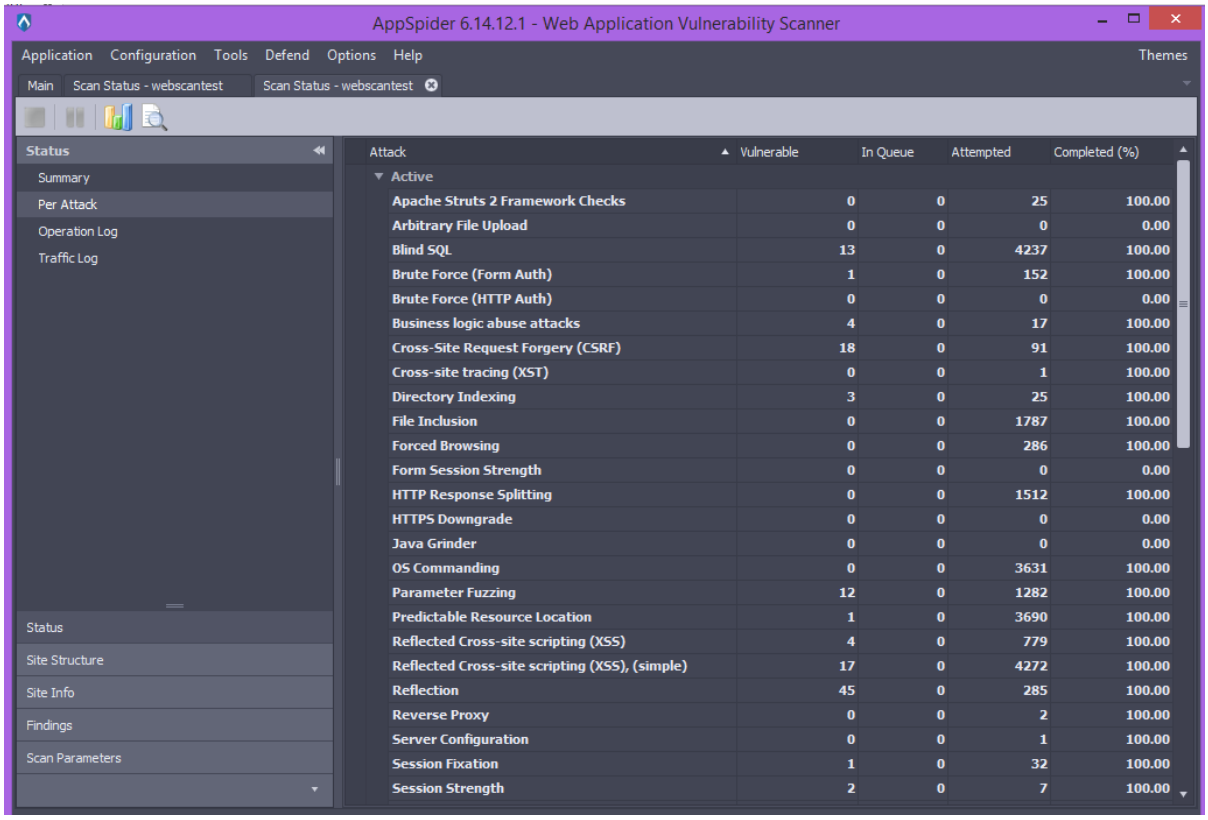
You may find the following information in the panel:

- Name of the scan
- Scan Information:
  - Current domain
  - Session/Form Authentication
  - Scan status (Running, Paused, Completed, Stopped)
- Start Time - Time and Date
- Time Elapsed
- Time Remaining
- Progress bars with scan progress as a percentage:
  - Overall progress
  - Scan Stage (Initialization, Scanning, Report generation)

- Scanning progress bars with counters:
  - Crawled (number of crawled links to number of queued links waiting to be crawled)
  - Attacked (number of attacks performed to number of queued attacks)
- Findings summary. Number of vulnerabilities: High, Medium, Low, Information, Safe.
- Network monitor:
  - Requests
  - Failed Requests
  - Request Delay (ms)
  - Speed (KB/s)
  - Response Time (ms)
- The events table is a list of events with date/time, severity icon and a description of the event.

Per Attack

This panel allows you to observe the list of attacks.

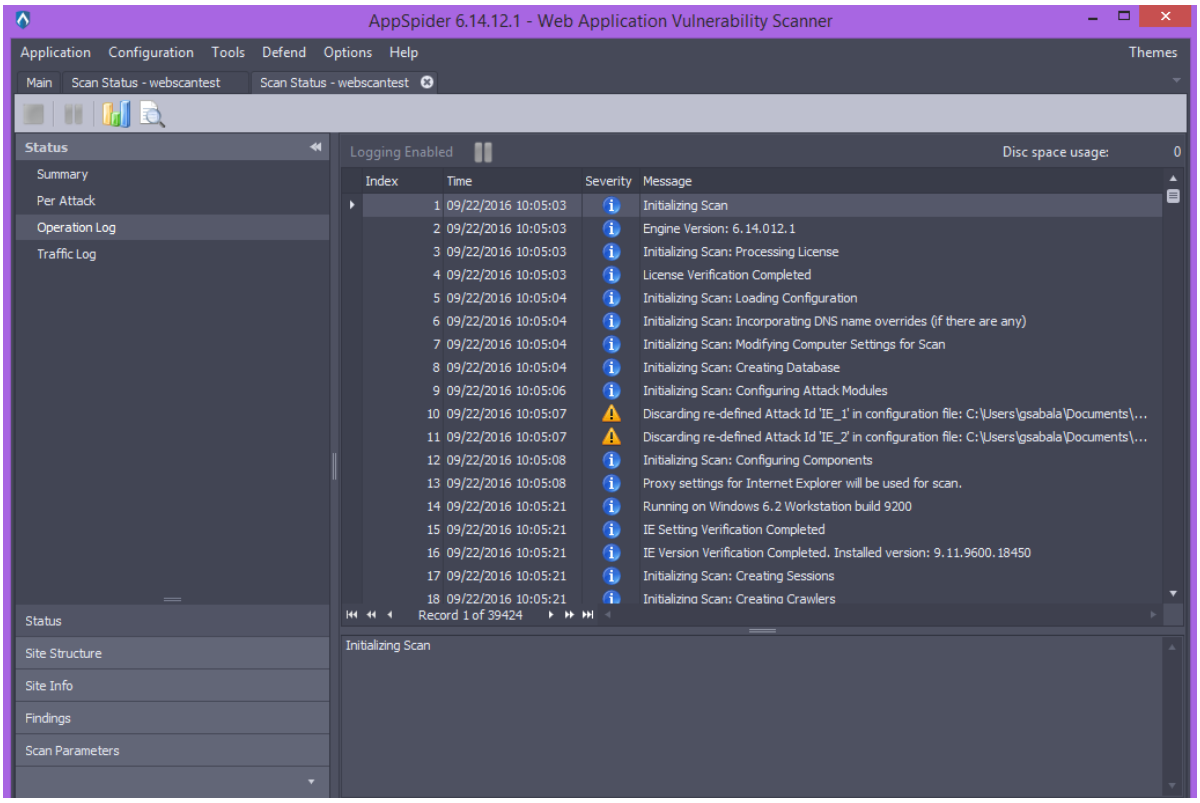


The table has two parts: *Active* and *Passive* attacks. You may collapse any part of the table. The following columns are presented:

- *Attack* - the name of the attack
- *Vulnerable* - the number of vulnerabilities found
- *In Queue* - the number of attacks of the current type in the queue waiting to be performed
- *Attempted* - the number of attacks attempted
- *Completed (%)* - the percent of attacks completed out of the total

Operation Log

This panel allows you to find the list of actions performed by AppSpider (e.g. Initializing the scan, Crawling, Performing an attack, Generating the report, etc.)



The *Logging enabled* button is displayed only when the scan is running:

- **Pause** - pauses logging
- **Start** - resumes logging

Disc space usage is presented on the screen.

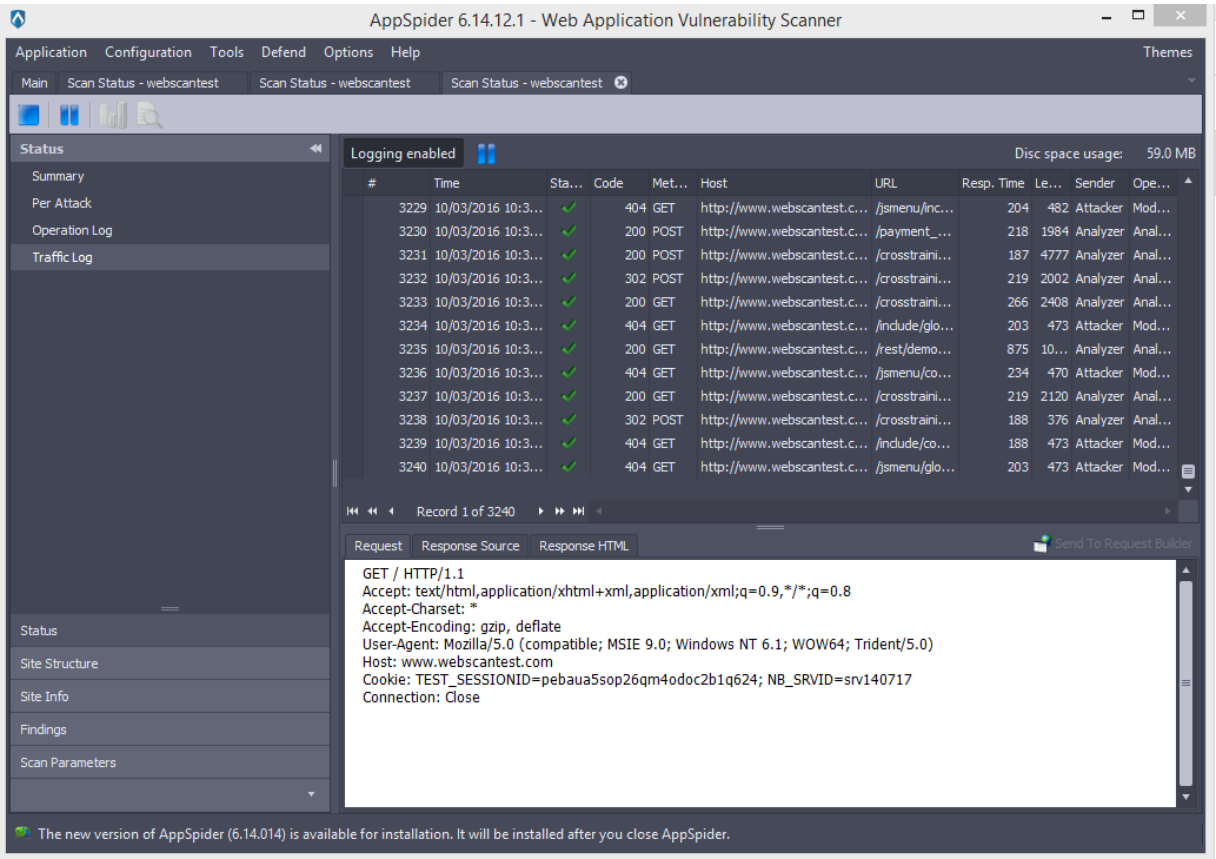
The table contains the following columns:

- *Index* - the number of the log entries
- *Time* - the time of the log entries
- *Severity* - the icon indicating the severity
- *Message* - event message

Selecting any row will display a message in the text box under the log table. The text box is locked and you will be unable to edit the message.

Traffic Log

This panel allows the user to find the list of requests performed by AppSpider.



The “Logging enabled” button is displayed only when the scan is running.

- **Pause**: pauses logging
- **Start**: resumes logging

Disk space usage is presented on the screen.

The table contains the following columns:

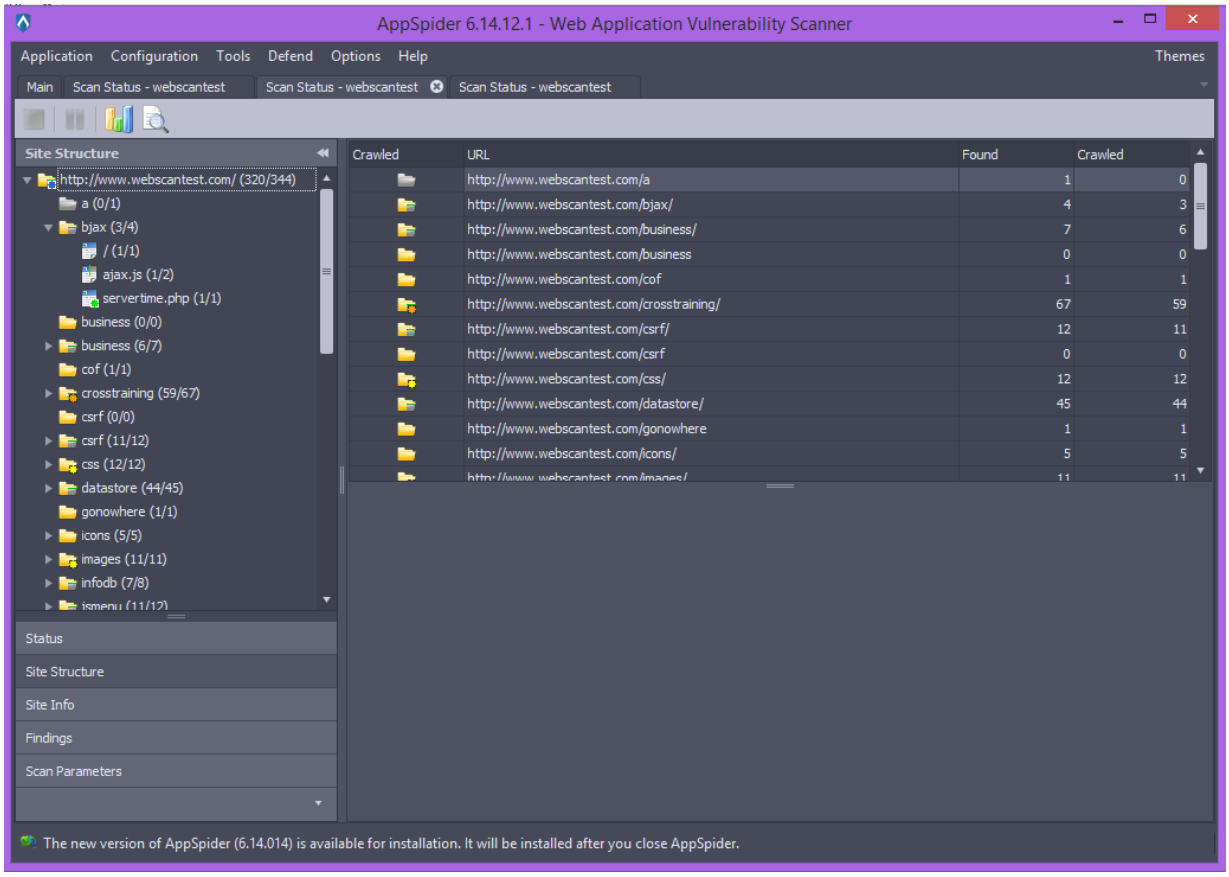
- *Index*: the number of the log entries
- *Time*: the time of the log entries
- *Status*: the status of the request
-  :Code = 0. Request failed. The server did not send any response
-  :Request succeeded. The response from the server was received.
- *Code*: response code
- *Host*: the host
- *Url*: the hosts relative URL
- *Resp. time*: the response time
- *Length*: the response length
- *Sender*: Indicates the component that initiated the request: Crawler, Login, Analyser, Attacker.
- *Operation*: the operation name

Selecting any row will display detailed information in the tabs under the table.

- *Request* - the request that was sent
- *Response Source* - the response source
- *Response HTML* - the response HTML

Site Structure

This panel allows you to observe the structure of the scanned site and review findings from the scan in the context of the overall structure of the site.



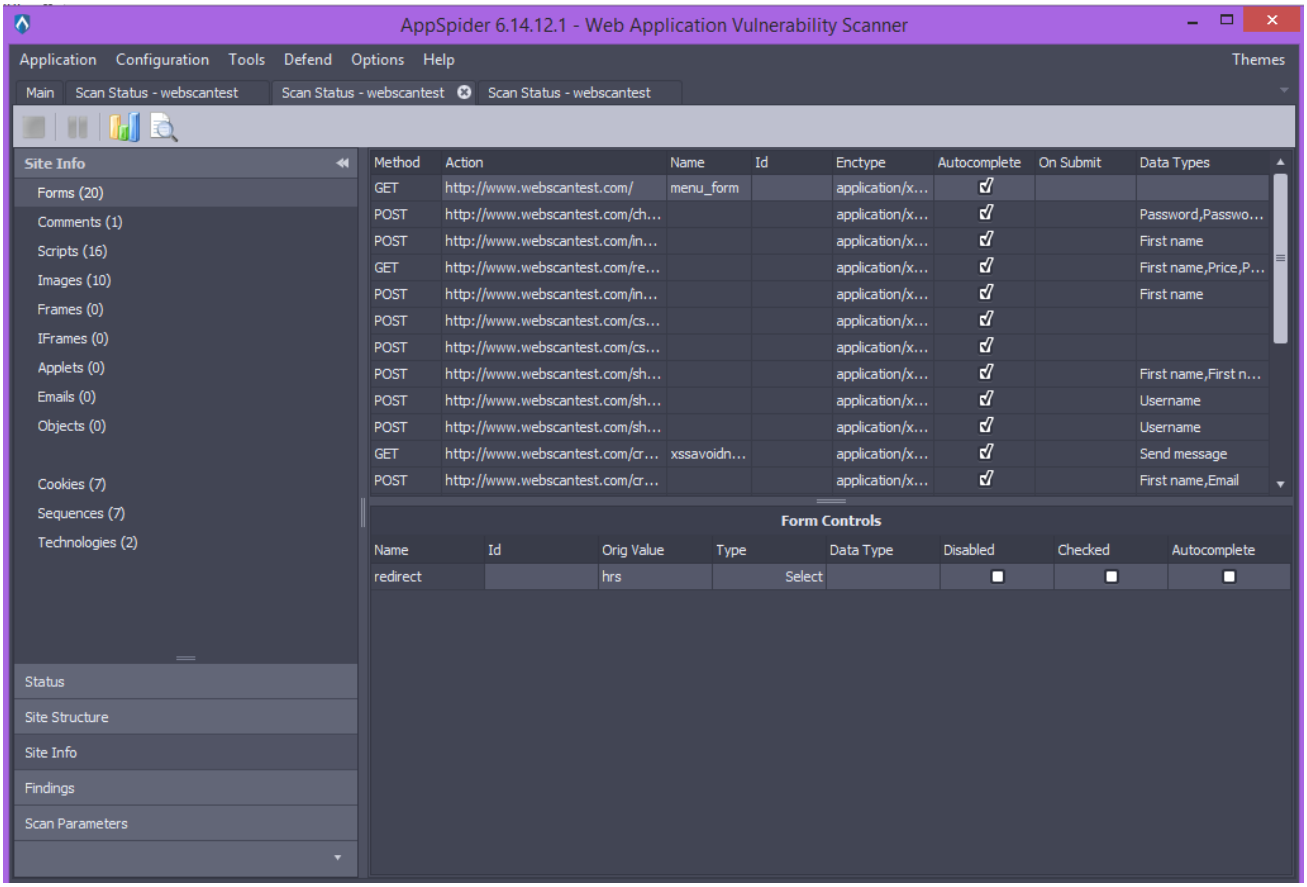
The Site Structure panel contains a tree structure of the crawled site.

After the node is selected its direct children nodes are displayed on the Site Structure table. The table contains following columns:

- *Crawled* - an icon (either file or folder) indicating the crawl status of the object:
    - Gray: not crawled
    - Partial Gray: partially crawled (folders)
    - Color: crawled
    - Dot (in corner): there is an associated finding; right-click to go to the finding. The dots are color-coded depending on the severity of the finding:
        - Red - High
        - Orange - Medium
        - Yellow - Low level finding
        - Blue - informational
        - Green - reflection analysis findings
- *Method* - request method (GET or POST)
- *URL* - the complete URL
- *Code* - the response code
- *Length* - the response length

Site Info

The *Site Info* panel allows you to find all the elements presented on the scanned site.



The left panel contains a list of elements found on the scanned site divided by type.

When you select any element type on the left, all elements of that type are displayed on the right side of the panel.

Selecting a row in the Main table populates the Details section with information about the row.

Every element type has its own list of parameters.

Forms

Main table:

- *Method* - request method (GET or POST)
- *Action* - the action link
- *Name* - the name of the form
- *Id* - the ID of the form
- *Enctype* - the encryption type
- *Autocomplete* - the autocomplete status for the element
- *On Submit* - the form OnSubmit script handler status of the element
- *Control Types* - types of controls(eg. Username, Password, Search etc.)

Details table *Form Controls* - the list of controls on the form:

- *Name* - the name of the element
- *Id* - the ID of the element
- *Orig Value* - the value of the element
- *Type* - the type of the element
- *Data type* - the data of type
- *Disabled* - marked when the element is disabled
- *Checked* - marked when the element is checked
- *Autocomplete* - marked when the element has autocomplete.

Comments

Main table:

- *Comments* - the first line of the comment
- *Details* - the text area with the comment

Scripts

Main table:

- *Source* - the URL of the source file.
- *Type* -Script type (javascript, vbscript, other values possible)
- *Deferred* - Indicates whether the deferred attribute was set on the script
- *Script* - the first line of the scrip, Inlined Script body (as opposed to external script)

Script table:

- Text area with the script

Images

Main table:

- *Source* - the URL of the image
- *Height* - the height of the image
- *Width* - the width of the image
- *Alt* - the alternative text of the image

Frames

Main table:

- *Source* - the URL of the frame
- *Long description* - the long description of the frame

IFrames

Main table:

- *Source* - the URL of the frame
- *Long description* - the long description of the frame

Applet

Main table:

- *Code* - the location of the code
- *Archive* - the archive attribute
- *Codebase* - codebase attribute

Email

The main table contains the list of emails.

Objects

Main table:

- *Codebase* - codebase attribute
- *Class Id* - classid attribute
- *Data* - data attribute
- *Use Map* - user map attribute
- *Archive* - archive attribute
- *Code Type* - code type attribute
- *Name* - name attribute

Cookies

Main table:

- *Name* - the name of the cookie
- *Protocol* - the application layer protocol
- *Domain* - the domain of the cookie
- *Path* - the path to the page setting the cookie
- *Value* - the value of the cookie
- *Secure* - marked when the cookie is secure.
- *Date* - cookie expiration date
- *Version* - the version attribute
- *Comment* - the comment attribute
- *Http Only* - marked when HTTP only is enabled

Sequences

Main table:

- *Name* - the sequence name
- *Steps* - the number of steps
- *Type* - the sequence type

Sequences Steps table:

- *Method* - the request method (GET or POST)
- *URL* - the complete URL
- *Code* - the response code
- *Length* - the response length
- *Transition Type* - the transition Type

Selecting any row will show detailed information in the tabs under the table.

- *Request* - the request that was sent
- *Response Source* - the response source
- *Response HTML* - the response HTML

Technologies

Main table:

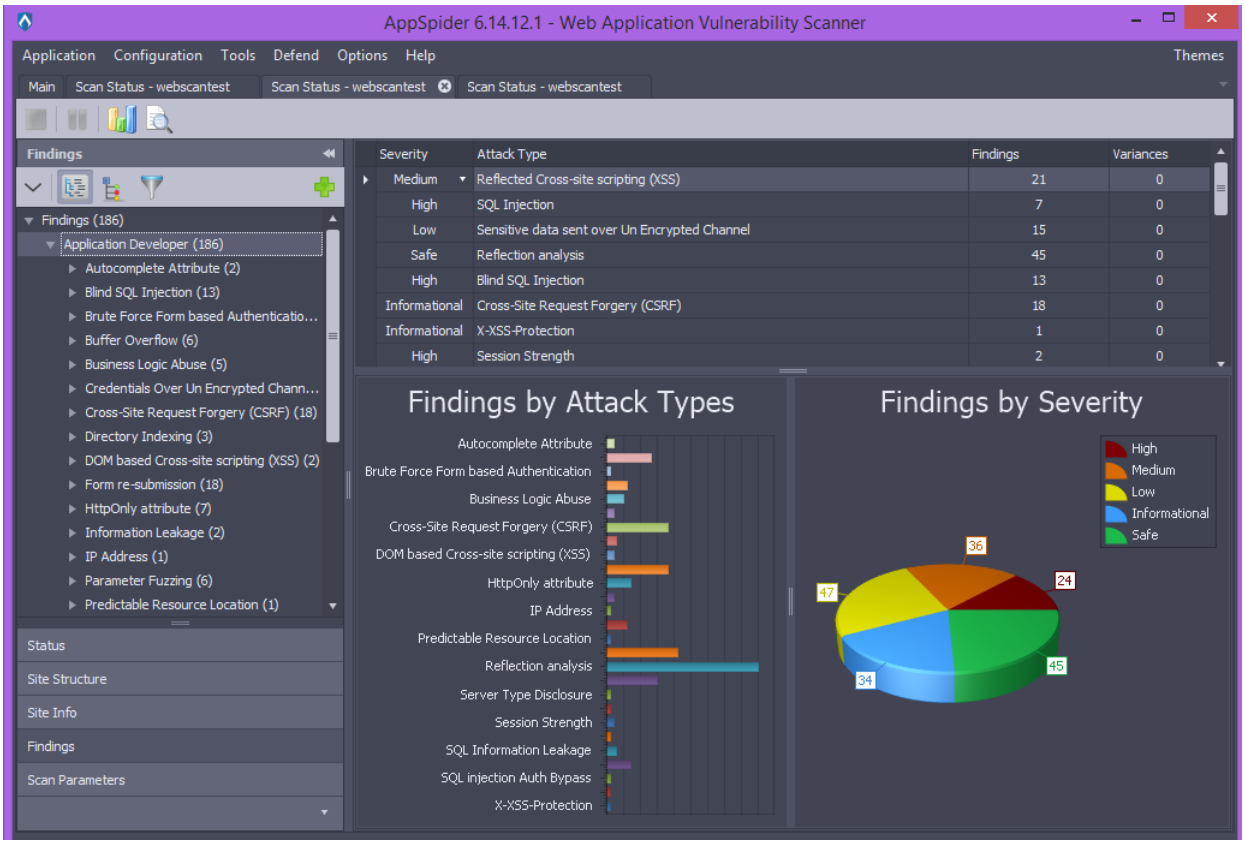
- *Name* - the technology name
- *Found* - number of found technologies
- *Crawled* - number of crawled technologies

Details table:

- *URL* - the technology URL
- *Found* - number of found technologies
- *Crawled* - number of crawled technologies

Findings

In the *Findings* panel, you can observe the vulnerabilities statistics, charts and vulnerabilities details.



The left panel has the tree view of the vulnerabilities with count in brackets.

You can click the “Expand all” button ( )to expand the entire vulnerability tree:

Findings

Attack class

Attack type

Finding location

Attack Variance

Vulnerabilities panel

The *Vulnerabilities* panel contains the statistics for all vulnerabilities:

- *Attack class* table:
  - Attack class - the name of the attack class
  - Findings - the number of vulnerabilities found
  - Variances - the number of attack variants
- *Findings by Attack Classes* chart
- *Findings by Severity* chart

Attack class panel

The *Attack class* panel contains the statistics for the selected attack class:

- Attack type table:
  - *Severity* - the severity of the attack type
  - *Attack type* - the name of the attack type
  - *Findings* - the number of vulnerabilities found
  - *Variances* - the number of attack variants
- *Findings by Attack Class* chart
- *Findings by Severity* chart

Attack type panel

Attack type panel contains the statistics for the selected attack type:

a. *Vulnerabilities for the Attack type* table - the user may select a vulnerability and observe the following options:

- *Severity* - the editable combo box with the vulnerability severity.

Once you have expanded the tree or a node, you can click on a vulnerability type to view vulnerabilities of that type. You can:

- Change the severity of the vulnerability
- Ignore the vulnerability using the **Ignore** checkbox
- Globally ignore using the **Globally Ignore** checkbox
- Mark the vulnerability as known using the **Known** checkbox
- Add notes using the **Notes** button
- Add new vulnerability using the **Add New** button.
- Filter issues by **All**, **Active**, **Ignore**.

The table contains following columns:

- *Method* - the request method
- *URL* - the finding URL
- *Parameter* - the vulnerable parameter
- *Variances* - the number of attack variants
- *Ignore* - the user marks this checkbox to ignore the issue
- *Globally Ignore* - the user marks this checkbox to globally ignore the issue
- *Known* - the user marks this checkbox to set issue status as Known
- *Notes* - displays notes about the vulnerability when clicked

Details

The following options are available in the *Details* tabs:

- *Description* - the description for the selected attack type.
- *Recommendation* - the recommendation for the attack type.
- *References* - the reference to the documentation about the attack type

Finding location

The *Finding location* tabs contain the information about the location of the finding.

You can:

- Ignore the vulnerability by using the **Ignore** checkbox
- Globally ignore it using the **Globally Ignore** checkbox
- Mark the vulnerability as known using the **Known** checkbox
- Add notes using the **User notes** text area
- Add a new vulnerability using the **Add New** button.

The *Finding location* table contains the following information:

- *Attack Type* - the vulnerability attack type
- *Attack Class* - the vulnerability attack class
- *Severity* - the vulnerability severity
- *Method* - the request method
- *URL* - the finding URL
- *Parameter* - the vulnerable parameter
- *User Notes* - displays notes about the vulnerability

Variances

The *Variances* table contains the following information about the attack variant:

- *URL* - the finding URL
- *Parameter* - the attack variant parameter
- *Ignore* - the user marks the checkbox to ignore the attack variant
- *Notes* - shows the note for this attack variant.

The Variance panel contains the information about the attack variant.

The following read only fields are available:

- *Attack Type*
- *Attack description*
- *Original Value*
- *Attack Value*
- *Vulnerability*

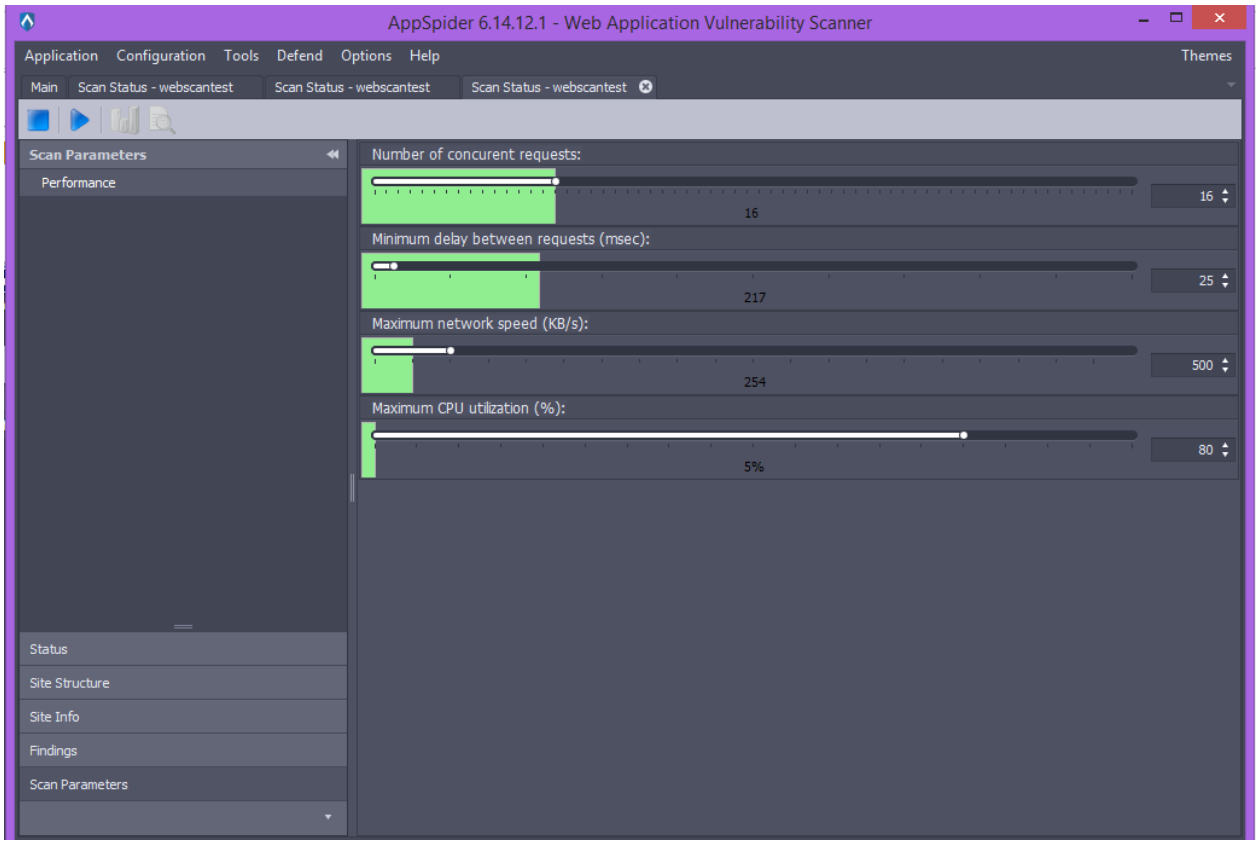
- The *Ignore* checkbox ignores the attack variant if marked.
- The *Notes* text area allows you to enter notes.

The following tabs are available for Original Traffic:

- *Request* - the web request
- *Response Source* - the web response
- *Response HTML* - the response view

Scan parameters

The panel allows you to observe the scan performance statistics.



The right panel has the four performance bars with counters:

- *Number of concurrent requests* should be between 1 and 64.
- *Minimum delay between requests* should be between 0 and 1000 milliseconds.
- *Maximum network speed* should be between 1 and 10 000 kilobyte per second.
- *Maximum CPU utilization* should be between 10 and 100 %.

Bulk Files Import

The Bulk Files Import allows you to import several pieces of data from the file directory.

The window contains the following options:

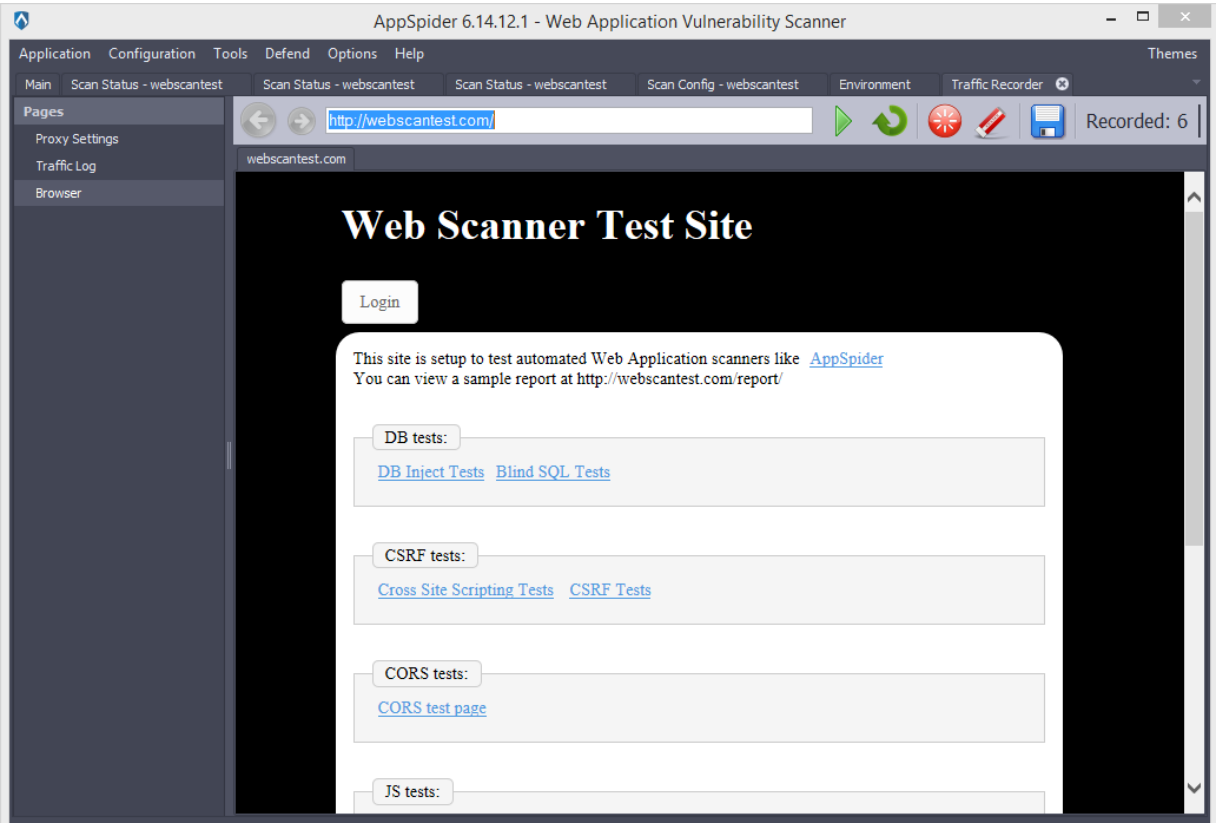
- *Folder*: opens file system for selecting a directory
- *Filter*: selection list for file extensions
- *Process internal folders recursively* checkbox
- *Progress* bar
- **Start** action button: runs the file upload process
- **Cancel** action button: closes the window without any changes

Traffic Recorder

The traffic recorder feature is used to record traffic for further scanning–for example, restrict a scan to recorded traffic.

The recorder can be started by pressing the **Record Traffic** button on the left side menu or by using **Tools** -> **Traffic Recorder** in the top menu.

Browser

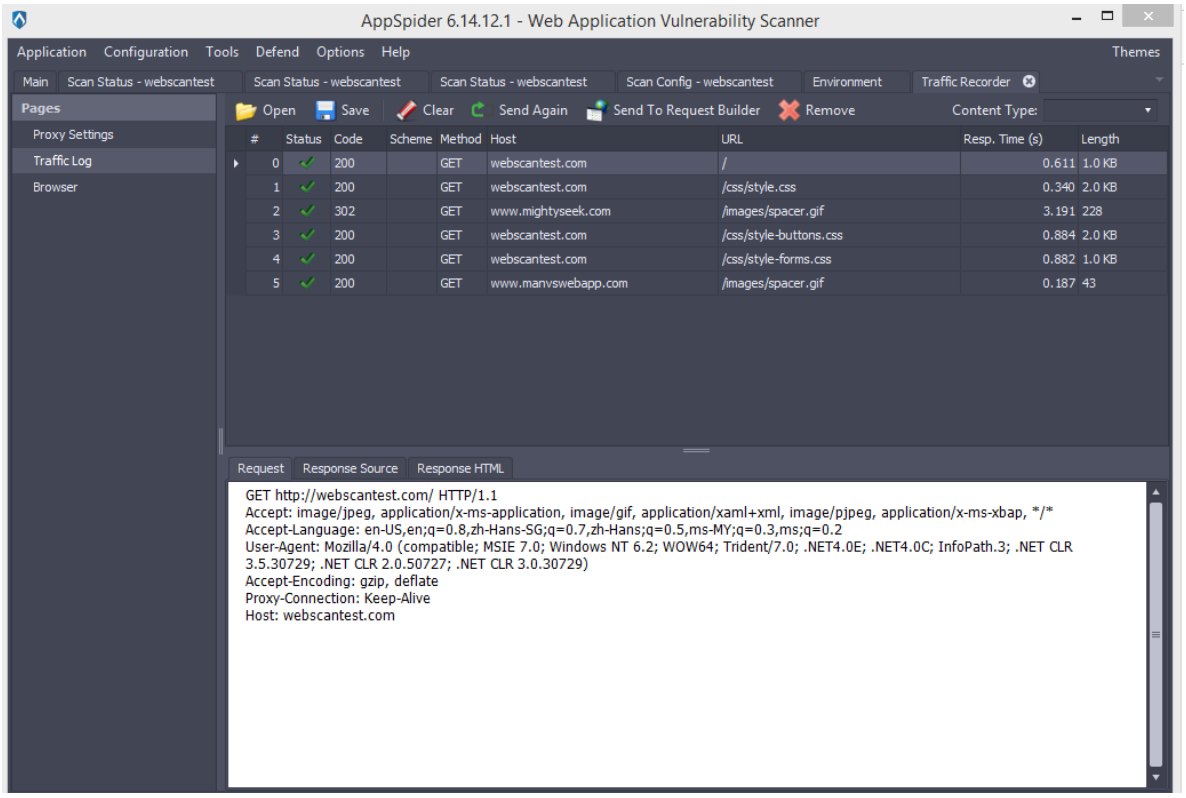


The browser control will be displayed by default when you start the Traffic Recorder.

You will find the following controls to manage the browser:

- **Back/Forward**: allows the you to navigate back and forward
- **Go**: the browser will open the specified URL
- **Refresh**: reload the page
- **Restart**: clear all saved traffic and start recording from the beginning
- **Clear cookies**: clear all cookies
- **Save**: save recorded traffic to a file

Traffic log



The Traffic Log panel displays recorded traffic for each request. You will find the following controls there:

- **Open**: loads the saved traffic file
- **Save**: saves recorded traffic
- **Clear**: clears the saved traffic
- **Send again**: sends the request once again (it will not be recorded in the traffic log)
- **Send to request builder**: opens the request in the *Request Builder* tab
- **Remove**: removes a selected request from the traffic file

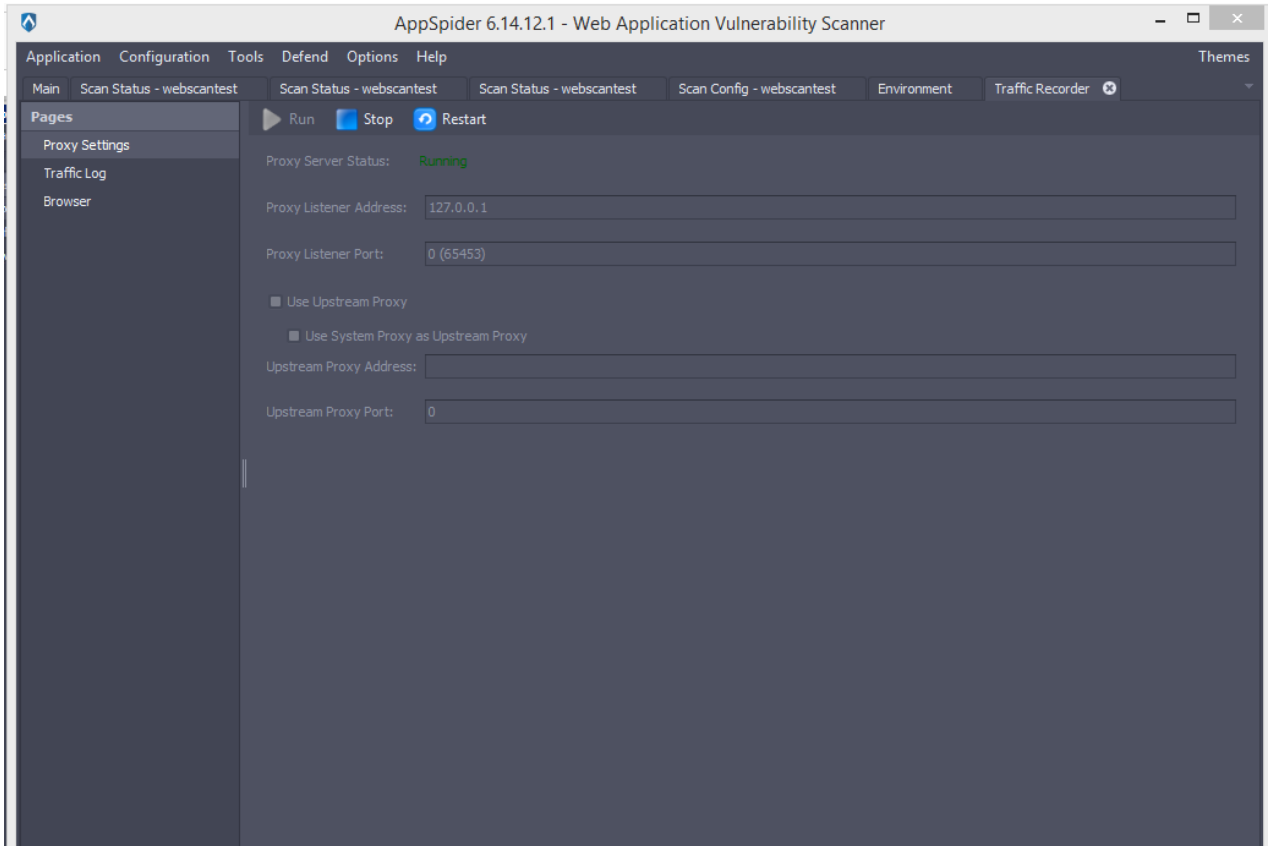
The table contains the following columns:

- *#* - the index number of the log entries
- *Status*: the status of the request:
  -  - Code = 0. Request failed. The server did not send any response
  -  - Request succeeded. The response from the server was received.
- *Code*: response code:
  - *Host*: the host
  - *Url*: the host's relative URL
  - *Resp. time*: the response time
  - *Length*: the response length

Selecting any row will display detailed information in the tabs under the table:

- *Request*: the request that was sent
- *Response Source*: the response source
- *Response HTML*: the response html

Proxy settings

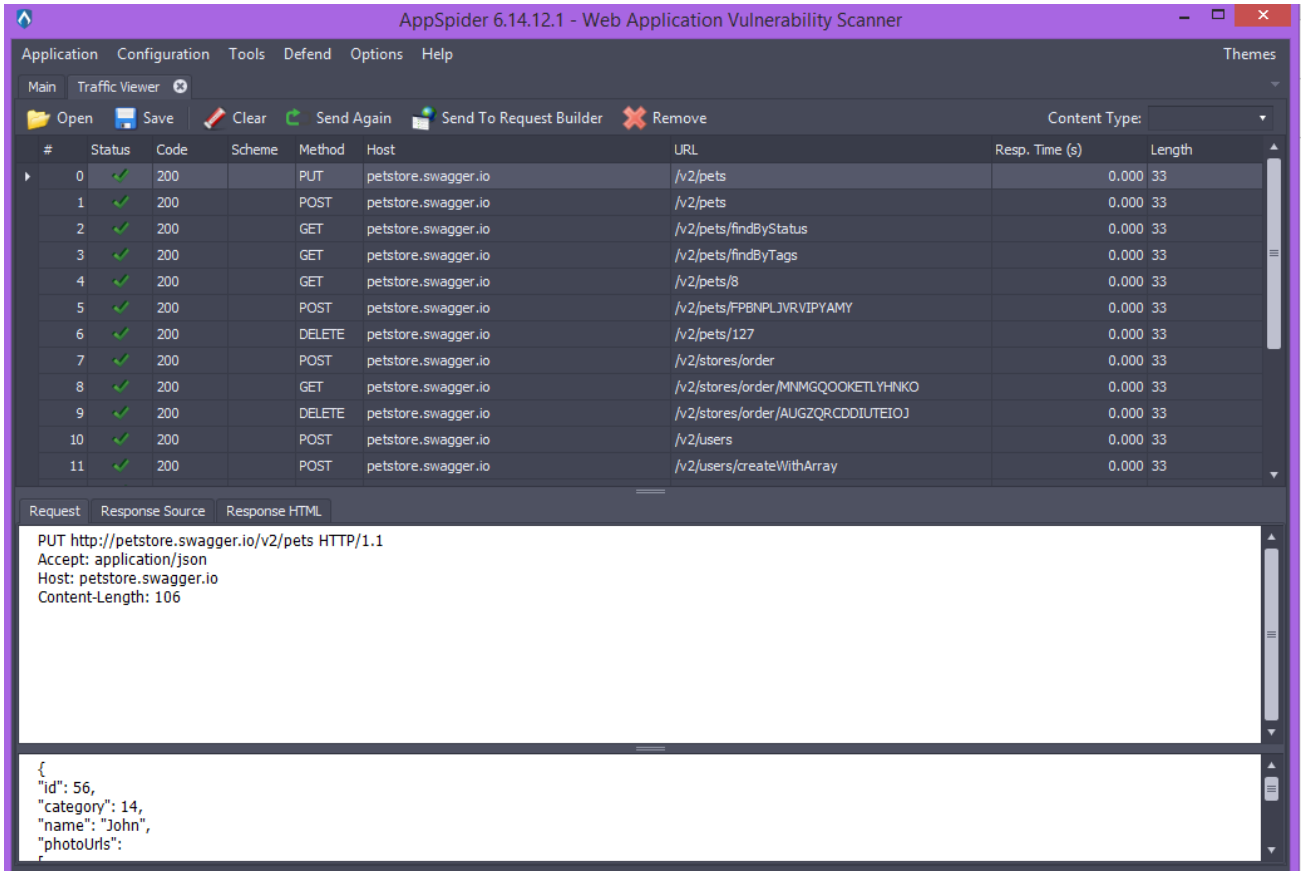


The Proxy settings panel allows you to manage the proxy port and IP address. By default, the proxy server uses the port assigned by the system; a zero (0) value in the “Proxy Listener Port” will be changed to the default value. The default value for *Proxy Listener Address* is 127.0.0.1, but you may change it to another accessible IP address in the system. If there is another proxy in the system,you may use the *Use upstream proxy* checkbox and provide the direct IP address and port values, or just check *Use system proxy as upstream proxy*.

Traffic Viewer

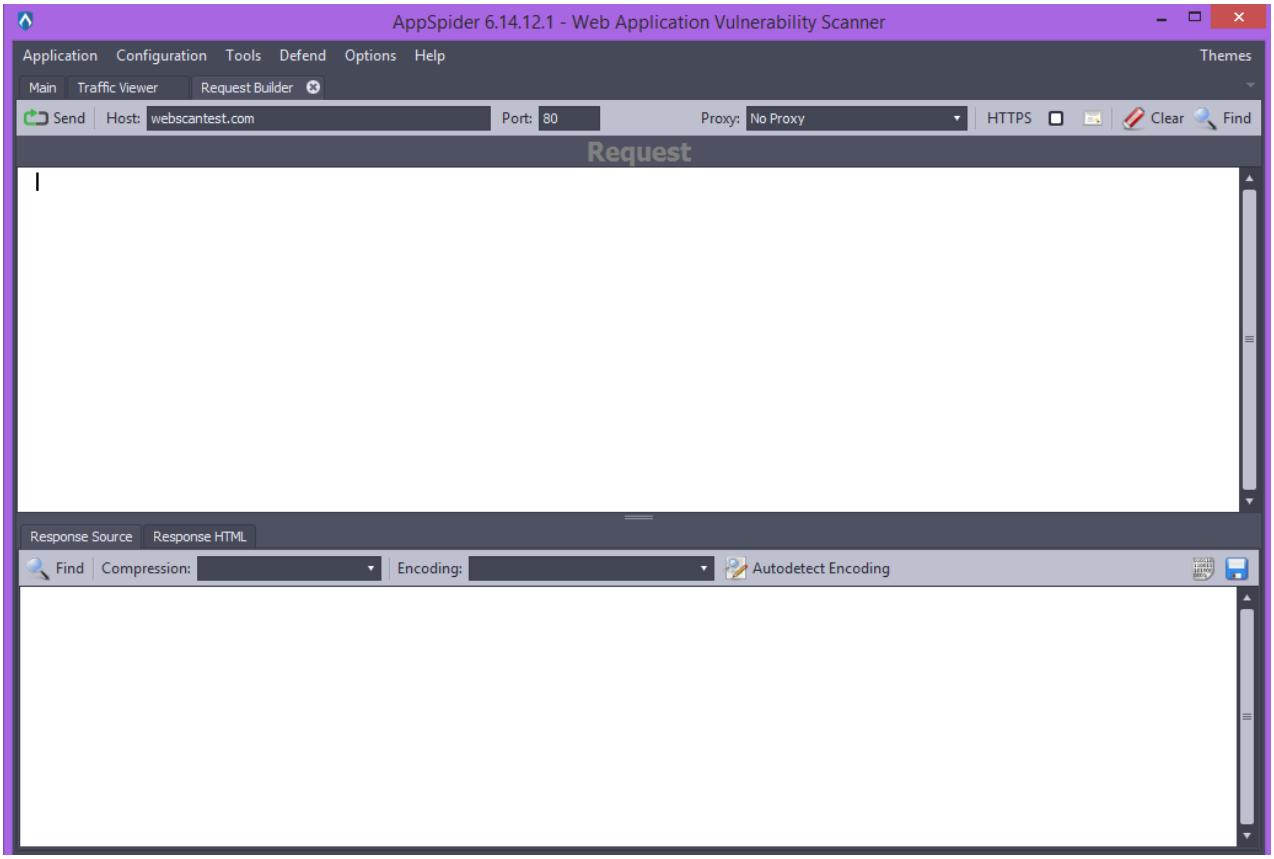
The Traffic Viewer allows you to import and manage a traffic file.

Once you have imported a file, the panel displays the same actions and data as the Traffic log panel of the Traffic Recorder.



Request Builder

The Request Builder allows you to create or modify a web request and execute it on the specified host. It allows you to explore possible attack vectors manually by experimenting with modifications to the request.You can view request syntax on a detail node of the *Findings* or *Site Structure* sections, and copy or send this syntax to the Request Builder. You can also open a new Request Builder window from the *Tools* menu.

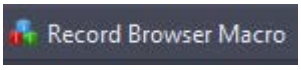


The Request Builder allows you to send a request (with modifications) in plain text format.

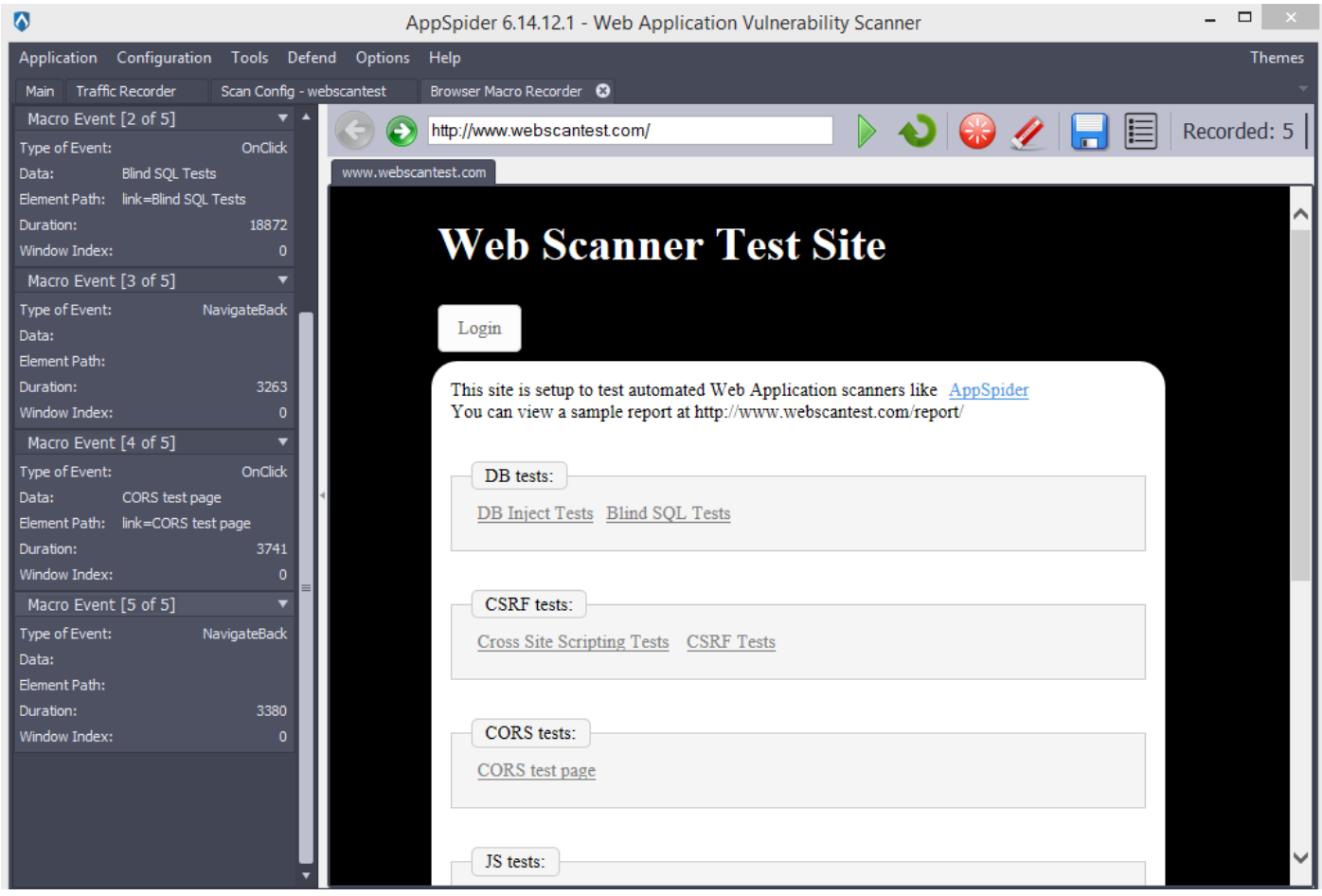
To execute the request, click the **Send** button - the http web request will be executed and the response text will be placed in the response text box.

When **Clear** is clicked, the request text box at the bottom will be cleared.

Browser Macro Recorder

You may record the macro using AppSpider's web browser by clicking on the 

button located in the Browser Macro page in the scan configuration .



The browser opens with the first URL from the URL list as the home page. All the user's actions will be recorded and used as a macro for crawling and attacks. You can use the following controls to manage the browser:

- **Back/Forward**: allows you to navigate backward and forward
- **Go**: the browser will open the specified URL
- **Refresh**: reload the page
- **Restart**: clear all saved traffic and start recording from the beginning
- **Clear cookies**: clear all cookies
- **Save**: save recorded traffic to the file
- **View**: opens macro event viewer

Regex Builder

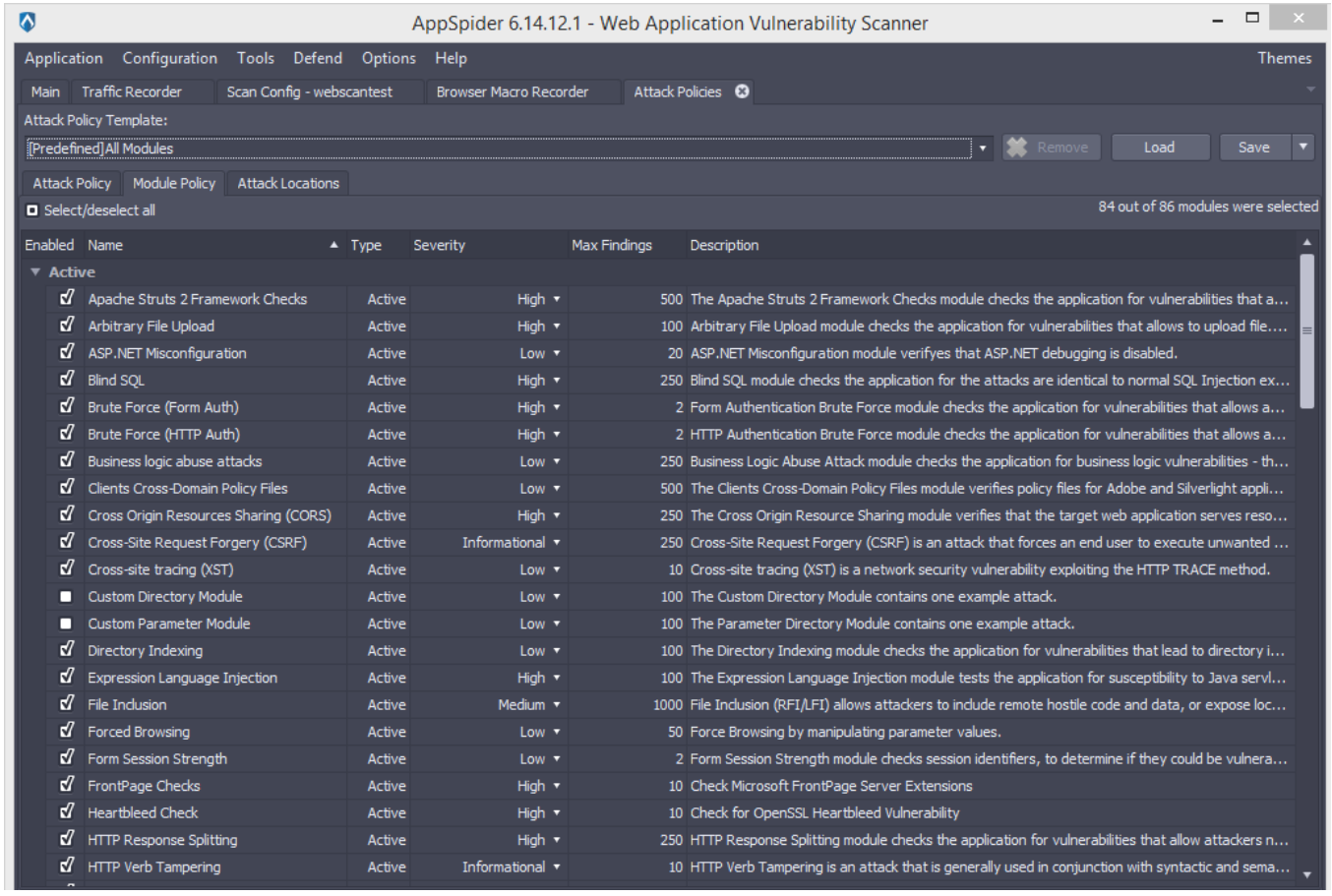
The Regex Builder panel allows you to create and test custom regular expressions. Regular expressions can be used to create patterns that pieces of text or code may match. Many settings in the application allow the option to specify a regular expression. By using the Regex Builder, you can experiment with a regular expression and make sure it performs as expected before using it in your scan configuration.

The panel contains the following fields:

- *Regex* text field with **Validate** action button
- *Regex analyzer*: displays whether the regular expression is valid or not
- *Sample text* text field: specify an example and test to see whether the expression can match it
- *Matches* results field
- *Validation result*: Sample view for Regex
- **Save** and **Cancel** action buttons

Attack policy

This panel allows you to customize the list of attack types, attack locations (such as directory, file, or path), and other properties.



Attack Policy Template list contains the types of predefined attack policies:

- **All Modules** - selects all modules.
- **Crawl Only** - clears all modules.
- **Passive analysis** - selects modules for passive analysis.
- **SQL Injection** - selects SQL Injection modules
- **XSS** - selects XSS modules
- **SQL Injection and XSS** - selects SQL Injection and XSS modules

You can save and load the custom/user attack policies using **Load** and **Save/Save as** action. buttons.

The user policies are located in the *AppSpider* folder under *AttackPolicies* folder. Each policy is located in a separate file.

If the policy name matches the scanner's default policy name, then the user's policy overwrites the scanner's default policy.

The user interface merges the default scanner policies with user policies in *Attack policy template* list.

You can click the **Save** button to save currently selected modules as a new policy.

AppSpider CMD

AppSpider has a console application that allows for the automation of creation of scan configurations and starting of scans.

The user needs to mark the AppSpider CMD checkbox during the installation process on the “Choose Components” window to install the program.

The following Help is available to the user:

AppSpider (c) 2015 Rapid7

-?, --help	Print help
sn, --scan-name	Set name of scan (and related directory)
-AM, --all-modules	Run all modules
-it, --include-traffic	Include HTTP traffic (uses a lot of storage)
-dnub, --do-not-use-browser	Do not use browser during crawling (for Web 2.0 sites)
-max, --max-links	Set max links to crawl
-nsbd, --no-sub-domains	No crawling of sub-domains (automatic constraints)
-nup, --no-unspecified-protocols	No crawling with unspecified protocols (automatic constraints).
-cnst, --add-constraint	Add constraint (overrides automatic constraints)
-h, --add-seed-url	Add to the URL set that seeds the crawler
-FS, --Form-Submission	Run Form Submission
-XSSS, --Cross-Site-Scripting-Simple	Run Cross-Site Scripting (XSS), Simple
-XSSR, --Cross-Site-Scripting-Reflected	Run Cross-Site Scripting (XSS), Reflected
-XSSD, --Cross-Site-Scripting-DOM	Run Cross-Site Scripting (XSS), DOM based
-SSL, --SSL-Strength	Run SSL Strength
-IL, --Information-Leakage	Run Information Leakage in error responses
-SCD, --Source-Code-Disclosure	Run Source Code Disclosure
-BL, --Business-Logic	Run Business logic attacks
-PF, --Parameter-Fuzzing	Run Parameter Fuzzing
-SCFG, --Server-Configuration	Run Server Configuration
-DI, --Directory-Indexing	Run Directory Indexing
-FSS, --Form-Session-Strength	Run Form Session Strength
-SC, --Script-Check	Run Script Check
-IDC, --Information-Disclosure-Comments	Run Information Disclosure in Comments
-IDR, --Information-Disclosure-Response	Run Information Disclosure in Response
-RP, --Reverse-	Run Reverse Proxy
UR, --Unvalidated-Redirect	Run Unvalidated Redirect

FB, --Forced-Browsing	Run Forced Browsing
-PRL, --Predictable-Resource-Location	Run Predictable Resource Location
-RFI, --Remote-File-Include	Run Remote File Include (RFI)
-SF, --Session-Fixation	Run Session Fixation
-WSPF, --Web-Service-Parameter-Fuzzing	Run Web Service Parameter Fuzzing
-BSQL, --Blind-SQL	Run Blind SQL
-FPC, --Form-Pattern-Check	Run Form Pattern Check
-AA, --Autocomplete-Attribute	Run Autocomplete Attribute
-CSRF, --Cross-Site-Request-Forgery	Run Cross-Site Request Forgery (CSRF)
-SQLA, --SQL-Injection-Auth-Bypass	Run SQL injection Auth Bypass
-AFU, --Arbitrary-File-Upload	Run Arbitrary File Upload
-SNCM, --Secure-Nonsecure-Content-Mix	Run Secure and non-secure content mix
-JG, --Java-Grinder	Run Java Grinder
-WB, --Web-Beacon	Run Web Beacon
-XST, --Cross-Site-Tracing	Run Cross-site tracing (XST)
-SQL, --SQL-Injection	Run SQL Injection
-BFFA, --Brute-Force-Form-Auth	Run Brute Force (Form Auth)
-BFHA, --Brute-Force-Http-Auth	Run Brute Force (HTTP Auth)
-SS, --Session-Strength	Run Session Strength
-HD, --HTTPS-Downgrade	Run HTTPS Downgrade
-P, --Profanity	Run Profanity
-ED, --Email-Disclosure	Run Email Disclosure
-URR, --URL-Rewriting	Run URL rewriting (Session IDs exposed in the URL)
-OSC, --OS-Commanding	Run OS Commanding
HRS, --HTTP-Response-Splitting	Run HTTP Response Splitting
-CA, --Cookie-Attributes	Run Cookie Attributes
-PD, --Privacy-Disclosure	Run Privacy Disclosure

Setting up to run Selenium scripts

AppSpider has the capability to run scripts for Selenium web browser automation programs and acquire URLs and cookies from the traffic generated by the scripts.

There are several possible ways to run Selenium scripts and set up the environment. Whereas, it is generally preferable for AppSpider to do as much legwork as possible to create and conduct a scan, this is not entirely feasible with Selenium due to the multitude of ways in which customers want to develop and run the scripts. AppSpider accommodates as many of these methods as possible but the burden of setting up Selenium rests with the customer.

For the purposes of this document, "AppSpider machine" indicates the computer on which AppSpider is installed and which will be running AppSpider Scans. Fundamentally, the script itself must exist on the AppSpider machine and the traffic that the script generates must proxy through the AppSpider machine on a port known to AppSpider. This port is 32768 by default but it can be configured in the scan configuration to be a different port. This proxy is necessary so AppSpider can scrutinize the links generated by the Selenium script and set cookies for the Selenium script and incorporate cookies generated by the Selenium script into its own scan context.

When configuring Selenium in AppSpider, make sure Proxy screen is set to **No Proxy**.

Technically, the script can be deployed to a different machine than the AppSpider machine and access it via a shared volume. However it is deployed, the AppSpider machine needs to be able to access it as a conventional filepath/filename and whatever Selenium environment is necessary to run the script must be present and active on the machine running the script (generally the AppSpider machine). AppSpider can be configured via the **Selenium Recordings** page of the scan configuration to run one or more of the following types of scripts:

- C# executable assemblies:

These are .exe files compiled from C# source invoking .NET Selenium libraries.

- Java files:

These are .jar or .class files compiled from Java source. Experience has shown that .jar files tend to work best. If shell executing the .jar file with no explicit parameters exectes the script in the desired way then it can simply be added as is. Otherwise, see Batch files below.

- HTML files:

These are the particular sort of HTML files that are output by the Firefox Selenium IDE. Internally they are parsed by AppSpider and run as if they were compiled as C# .NET Selenium scripts

instantiating an HtmlUnit RemoteWebDriver proxied through localhost:SeleniumPort. Per above, SeleniumPort is 32768 by default but is configurable. Due to the manner in which these are run, the Selenium server must be running on the AppSpider machine for the entirety of the scan and the .NET environment necessary to run .NET Selenium scripts must be installed on the AppSpider machine.

- Batch files:

These are .bat files and serve as a catch-all for miscellaneous ways of running Selenium scripts that do not fit with any of the above. This could be Ruby or Python, or a .jar file that requires particular parameters, etc. Whatever environment is necessary to successfully execute the .bat file standalone must be present and active on the AppSpider machine for the entirety of the scan.

Here are the standard steps for setting up a scan to run a Selenium script on the AppSpider machine:

1. Modify the Selenium script/scripts to proxy itself/themselves through the AppSpider machine.
    a. For example, If the Selenium script is instantiating its WebDriver object on the same machine on which it is running (AppSpider machine) then this will be 127.0.0.1:32768. The port is configurable in the scan configuratio).
    b. If the Selenium script is running on the AppSpider machine but instantiating a RemoteWebDriver on another machine, then, in addition to whatever Selenium environment necessary existing on that machine, the script code must proxy itself through the AppSpider machine and the AppSpider machine IP address must be routable from the RemoteWebDriver machine to the AppSpider machine. See the code sample below for an illustration of setting up the Selenium script proxy.
2. Configure the scan with the **Selenium Recordings** scan configuration page in AppSpider. Add as many scripts as you want executed.
    a. Note that they will be executed in order, so make sure to specify an order that does not create any dependency or invalidation issues.
3. Execute the Selenium server on the AppSpider machine.
    a. For example, `java -jar selenium-server-standalone-2.xx.0.jar`
4. If any other preparation specific to the particular installation of Selenium on the AppSpider machine or to the manner in which the scripts are to be run is necessary, do that here.
5. Run the scan.

Here is a C# code sample indicating how to set up a RemoteWebDriver object to proxy on the appointed port:

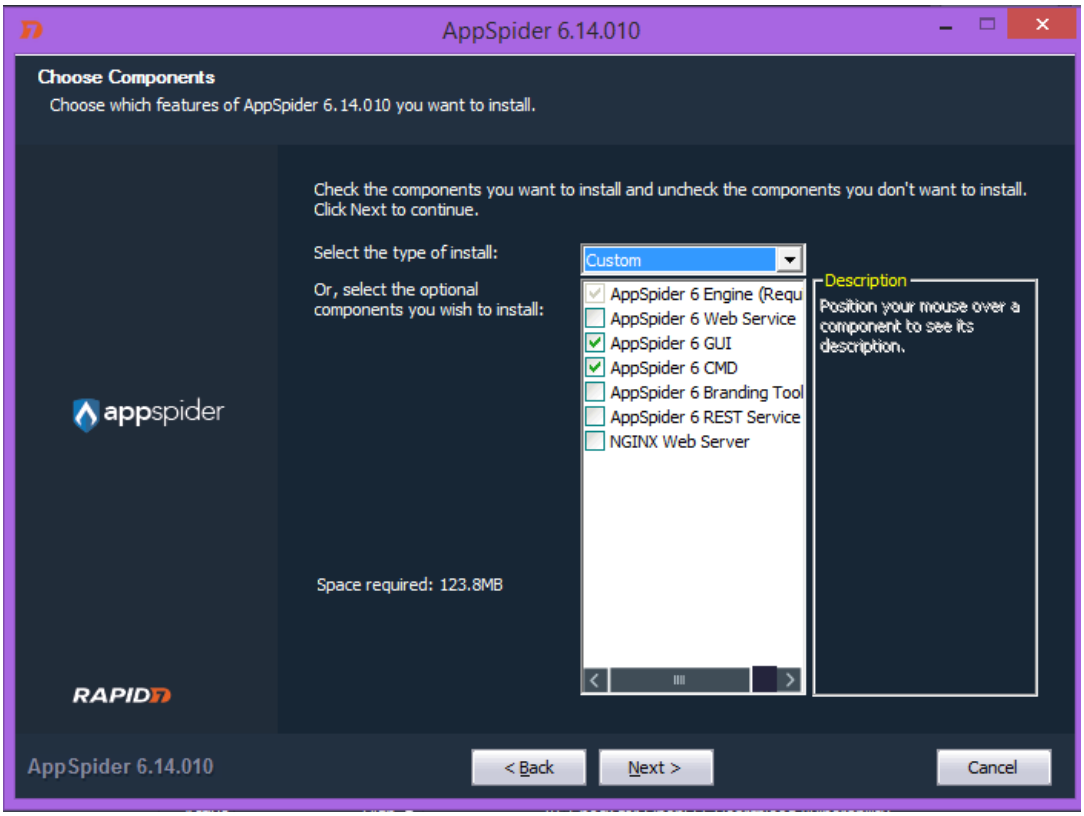
```
Proxy objProxy = new Proxy();
```

```
DesiredCapabilities  objDesiredCapabilities = DesiredCapabilities.Firefox
();

// If the RemoteWebDriver is to be instantiated on a different machine,
then instead of 127.0.0.1, put the IP address of the AppSpider machine
here.

objProxy.HttpProxy = "127.0.0.1:32768";

objDesiredCapabilities.SetCapability(CapabilityType.Proxy, objProxy);

// If the RemoteWebDriver is to be instantiated on a different machine,
then instead of 127.0.0.1, put the IP address of that machine in the URI.

using (IWebDriver objWebDriver = new RemoteWebDriver(new Uri
("http://127.0.0.1:4444/wd/hub "), objDesiredCapabilities))

{   // Main script code goes here.
```

AppSpider Web Service

To install the web service, select the **AppSpider Web Service** checkbox during the installation process in the *Choose Components* window. To learn more, see *AppSpider System Requirements* on page 5.

The web service provides a web interface to the scan engine that is used by AppSpider Enterprise. If AppSpider Enterprise is not being installed then the AppSpider Web Service should not be selected at install.



Rapid7 Branding Tool

The reports generated by the scanner upon scan completion contain images that brand them as a product of Rapid7. Certain clients have expressed interest in being able to generate reports with their own company's branding, such as when pen-testers use the tool and then want to deliver reports with their company's logos and color scheme. The Rapid7BrandingTool.exe was written to assist customers who want branded reports with generating the required images and encoding them into the destination files.

The two files that need to be generated are: BrandedImages.zap and ReportConstants.txt.

BrandedImages.zap contains all the static images (that is, not graphs and such that are dynamically generated) with a few key images replaced with the desired branded images.

ReportConstants.txt contains strings that show up in the footer of each report page, the title bar/tab of each page, and the company name in the disclaimer in the compliance reports.

These files need to be placed in the "My Documents\AppSpider" directory in order for the report generator to recognize and use them.

Rapid7BrandingTool.exe can be used to generate these files. The main/single screen walks you through the process of creating the images and the information for the text file, showing you a rough mockup of where they wind up in the report. If you have already created the images, click "Choose Directory" to point the tool to the directory that contains them. If you have created the ReportConstants.txt file already, the tool will sense that and populate the edit boxes on the bottom left with the contents. If you have not created the images already, you can still Choose Directory first or click "Create Images" which will prompt you to choose a directory if you have not already.

Whether you "Create Images" or "Choose Directory" with images already there, the listbox under those buttons will populate with the images, all initially selected. You can select and unselect individual images and then watch the mockup graphic (it takes a second or so) to see where each image shows up in the report pages. Whatever is selected at the time you click "Generate Files" is what will be placed in the BrandedImages.zap file. You can also type into the edits in the lower left and similarly see where that information shows up in the mockup graphic. "Create Images" creates all the image files that replace the standard image files with your branded/logo images. Of course we have no idea what images you want so these initial images are deliberately rudimentary patterns so they are easy to see in the mockup and so you can see what dimensions the images need to be as well as what the filenames are. You can simply edit these images directly or have your graphic artists create others following the same dimensions and copy them over but please note if you do the latter, the images MUST be .png (i.e. not .jpg, .gif, .bmp, etc.) and MUST have the same filenames. The dimensions can vary slightly but it is highly recommended that you retain at least the height of any image whose dimensions you change.

If you or your employees have web development knowledge, you can get more elaborate with the image edits including editing the Report.css style sheet to change alignment, justification, etc. of the images. You might choose to change a larger subset of the images than the standard case. When you are satisfied with all the images and possibly Report.css that you want to replace in this case, place the files (and only the files) into a directory and then point Rapid7BrandingTool.exe at that directory with the “Choose Directory” button.

So now you have created the images, either by using the “Create Images” button to create starting point images or by editing them from scratch. Now you will likely want to edit the text fields so that it does not say “Rapid7” at the bottom of each report page and elsewhere. The first time you use the tool, it is recommended that you type your company name into the edit box next to the “Populate” button and then click the button. This will populate 3 of the 4 edits with sample information predicated on your company name which you can then fine tune if you wish. The one edit that does not get populated is “Title App Name.” This is the name you are using to OEM AppSpider where applicable or it can be, “your company - Pentest results,” for example, or you may simply leave it blank. The text shows up in the title bar or tab in tabbed browsers.

Once you are satisfied with the images and the text fields, click “Generate Files” and the files BrandedImages.zap and ReportConstants.txt will be placed in the directory you selected with “Choose Directory.” Please note that “Generate Files” will incorporate any .css, .png, .policy, and .jar files selected in the listbox into the BrandedImages.zap file and they will therefore show up in the images directory of any report generated once BrandedImages.zap is placed in the “My Documents\AppSpider” directory and that is why it is important not to have any stray files of those types in the directory as they too will show up in the listbox cluttering it and making it confusing. As previously indicated, once you have BrandedImages.zap and ReportConstants.txt, copy them into the “My Documents\AppSpider” directory to have them show up in any reports generated from that point forward.

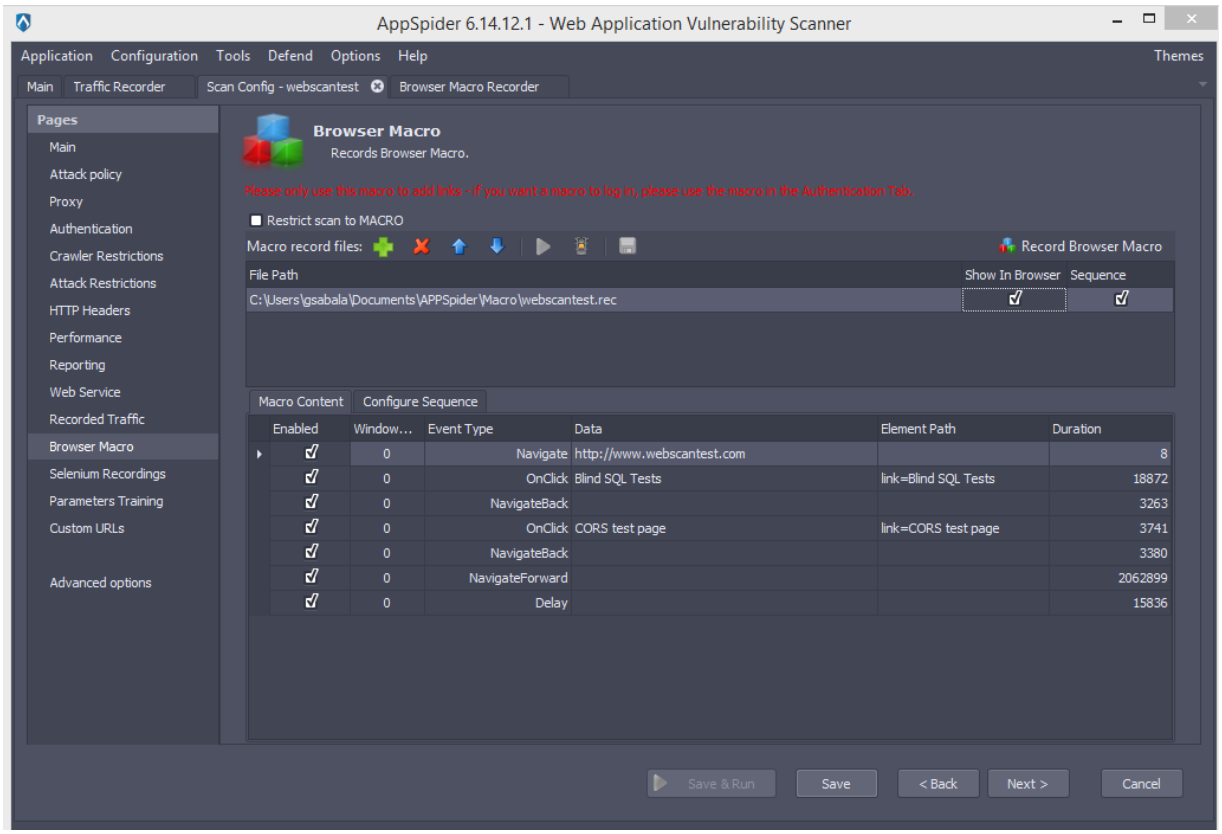
Sequences

Sequences allow users to create a sequence of requests that perform an action on the website. The typical sequence would be a shopping cart sequence, where a user first selects an item to buy, then adds it to the shopping cart, then proceeds to the checkout, enters a home address, payment option, and finally submits all the collected data to the web site to process the purchase of the item.

Usually, the sequence can not be interrupted, or jumped into in the middle - the website will throw an error if the request is done out of sequence. In order to test the sequence properly for security issues, the entire sequence needs to be replayed with every attack.

AppSpider has added support for attacking sequences. In order to attack a sequence as a whole, the user needs to start a macro recording, declare it as a sequence, and record it in the embedded in UI player to verify proper recording. AppSpider will run all its attacks with this sequence. Running attacks within a sequence can be slow, so it is advisable to limit the scan to this sequence only by selecting the **Restrict scan to Macro checkbox** on the Browser Macro config page.

By default, the embedded browser(s) are hidden. If you wish to observe the sequence attacking during the scan, then you will need to select the **Show in browser** checkbox.


AppSpider 6.14.12.1 - Web Application Vulnerability Scanner
Application Configuration Tools Defend Options Help
Themes
Main
Traffic Recorder
Scan Config - webscantest
Browser Macro Recorder
Pages
Main
Attack policy
Proxy
Authentication
Crawler Restrictions
Attack Restrictions
HTTP Headers
Performance
Reporting
Web Service
Recorded Traffic
Browser Macro
Selenium Recordings
Parameters Training
Custom URLs
Advanced options
Browser Macro
Records Browser Macro.
Please only use this macro to add links - if you want a macro to log in, please use the macro in the Authentication Tab.
Restrict scan to MACRO
Macro record files:
Record Browser Macro
File Path
Show In Browser
Sequence
C:\Users\gsabala\Documents\APPSpider\Macro\webscantest.rec
Macro Content
Configure Sequence
Enabled
Window...
Event Type
Data
Element Path
Duration
0 Navigate http://www.webscantest.com 8
0 OnClick Blind SQL Tests link=Blind SQL Tests 18872
0 NavigateBack 3263
0 OnClick CORS test page link=CORS test page 3741
0 NavigateBack 3380
0 NavigateForward 2062899
0 Delay 15836
Save & Run
Save
< Back
Next >
Cancel

Defend Introduction

Web Application Firewalls (WAFs), Intrusion Prevention Systems(IPS), and Intrusion Detection systems(IDS) are a highly recommended element of a security defense strategy but they have limitations. Out of the box rules often miss certain attacks.

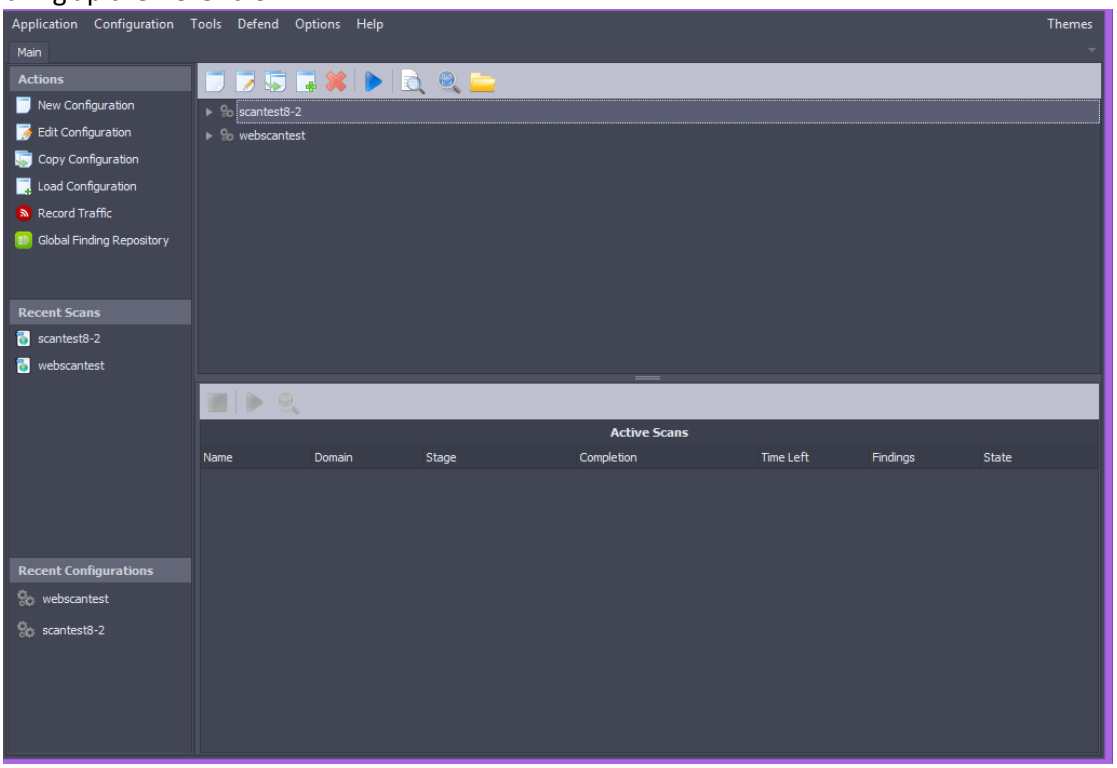
Using innovative automated rule generation, Defend, part of the AppSpider Pro, helps security professionals to patch web application vulnerabilities with custom rules in a matter of minutes instead of the days or weeks it can take by hand.

Without the need to build a custom rule for a WAF or IPS or the need to deliver a source code patch, Defend allows developers the time to identify the root cause of the problem and fix it in the code.

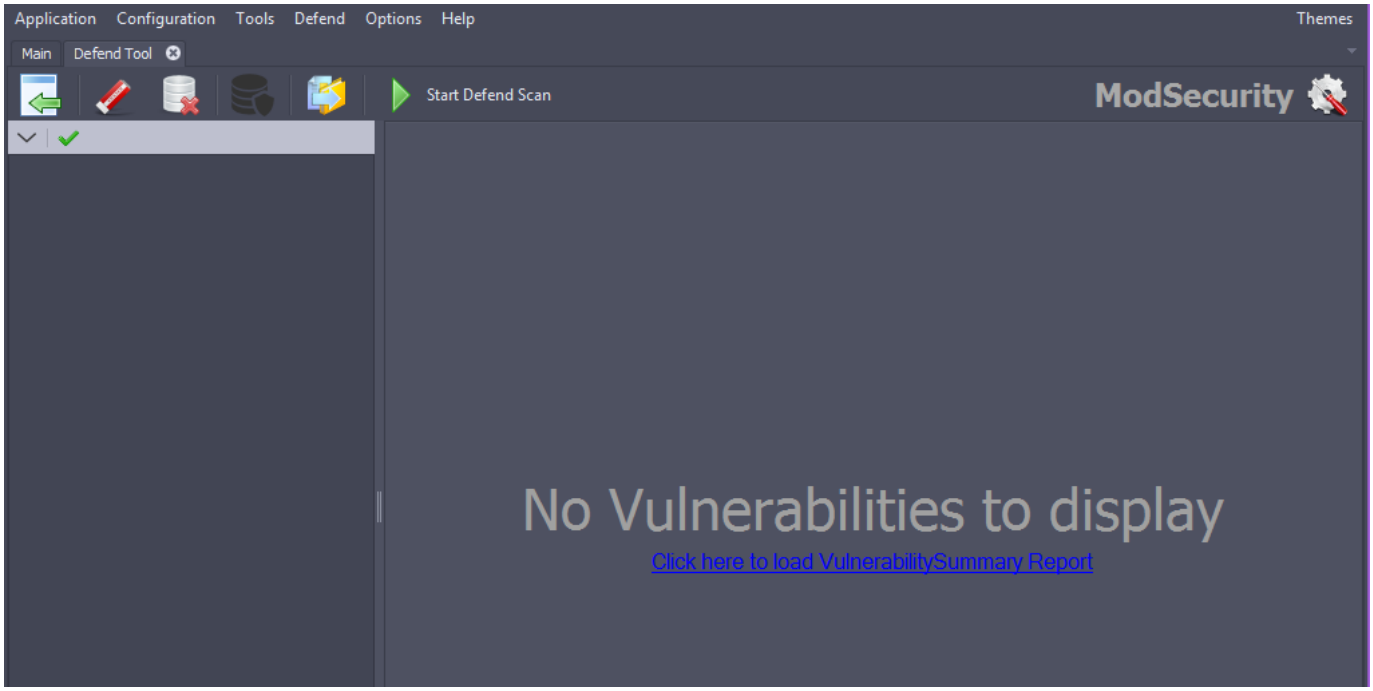
Defend allows you to easily create custom defenses for Web Application Firewalls(WAFs), Intrusion Protection Systems(IPS), or Intrusion Detection Systems(IDS), based on the results of vulnerability scans conducted with AppSpider Pro.

Defend User Interface:

On the top level menu next to the Tools section you will find the Defend menu which when clicked will bring up the Defend UI.



Once the Defend menu item is clicked you will see the Defend UI below.



Hovering your cursor over the icons will display a description of what function each icon will accomplish.

 - Load Finding

 - Clear Findings – Removes AppSpider Vulnerabilities Summary data.

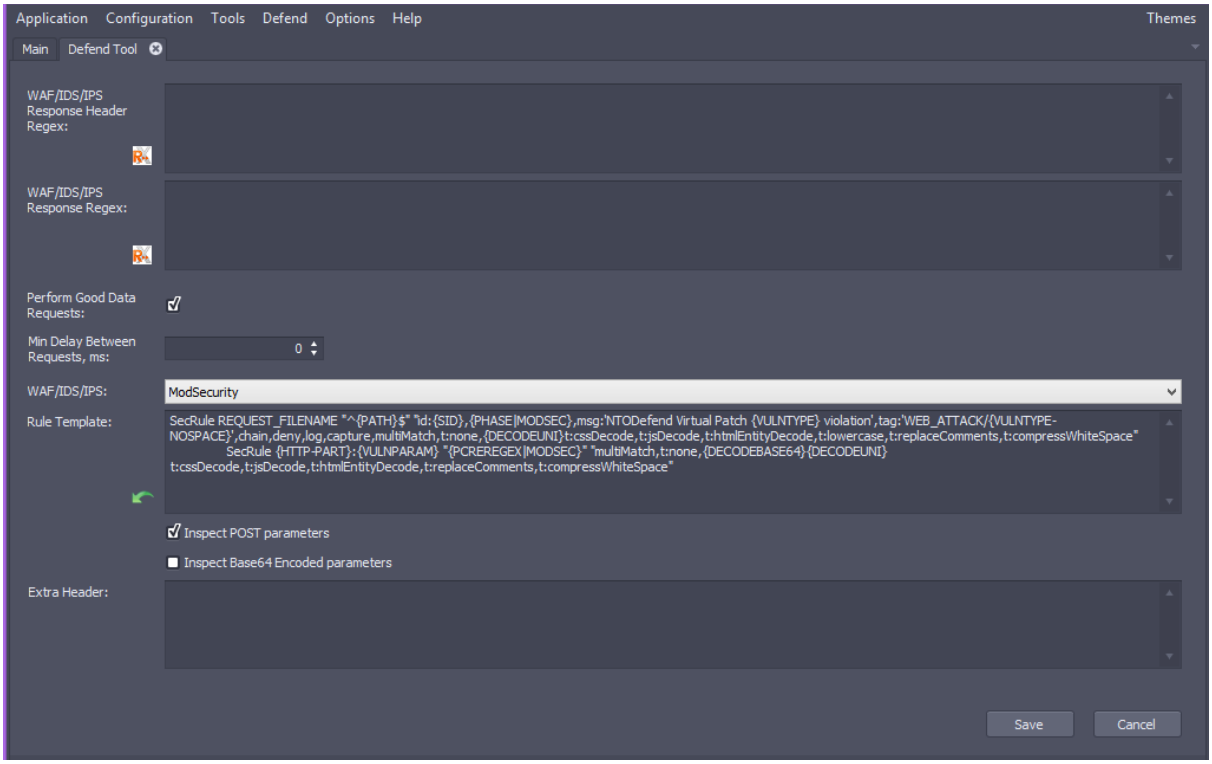
 - Clear DB – deletes vulnerabilities and good data.

 - Good data UI - Used to input/modify good data used for Defend Scans.

 - Export Rules – (default is Mod Security): export custom rules generated by Defend.

- Start a Defend Scan to confirm the selected vulnerabilities are no longer exploitable once the Defend rules have been loaded into your WAF

 - Options section which opens the UI listed in the next section.



The following elements are presented in this Options UI window:

- WAF/IDS/IPS Response Header Regex: editable field for the regular expression Defend uses to search values in headers.
- Response Regex: editable field for the regular expression Defend uses to search values in responses and exceptions in the processing of responses.
- Perform Good Data Requests: check box is selected by default.
- Min.Delay Between Requests, ms: 0 millisecond delay is predefined.
- WAF/IDS/IPS: select the WAF/IDS/IPS that you want to configure with Defend. The current supported WAF/IDS/IPS's are the following: ModSecurity, SourceFire/Snort, Nitro/Snort, Imperva, Secui/Snort, Akamai, Barracuda, F5, and DenyAll.
- Rule Template: the set of rules Defend uses with your WAF/IDS/IPS.
- Extra header: append extra headers to requests sent during simple scan.

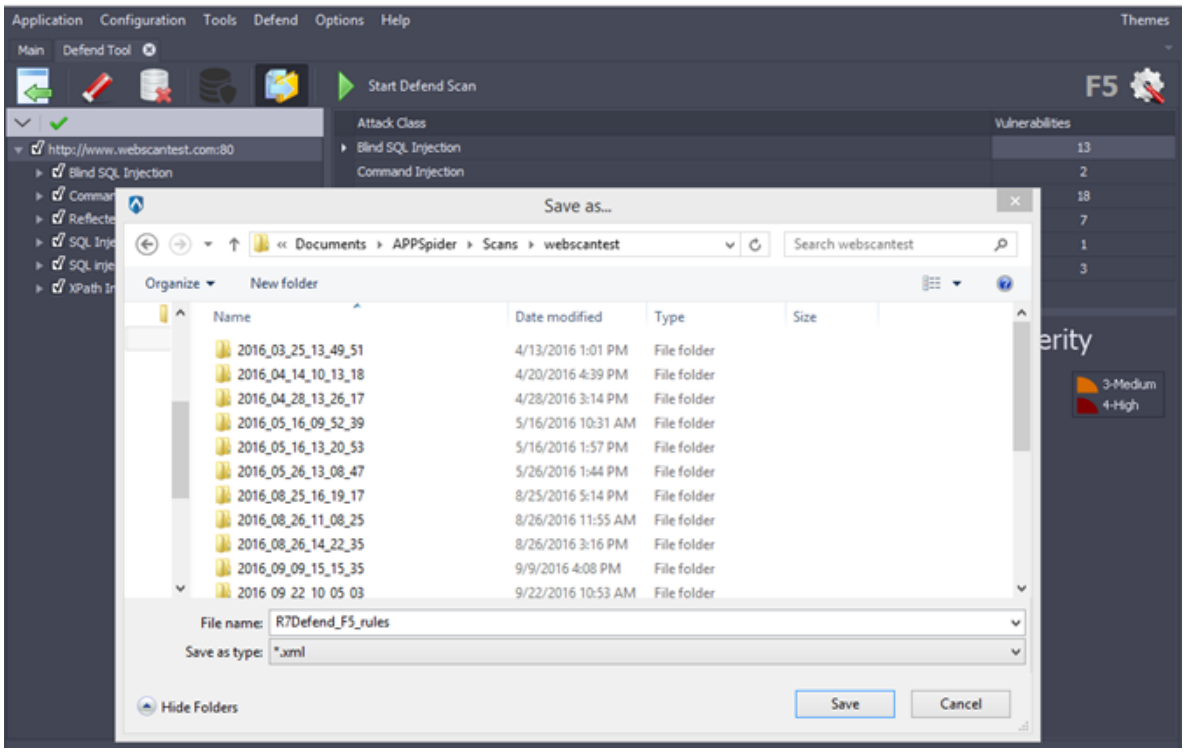
Generate Filters

Use your knowledge of the WAF/IDS/IPS and the application to create custom rules.

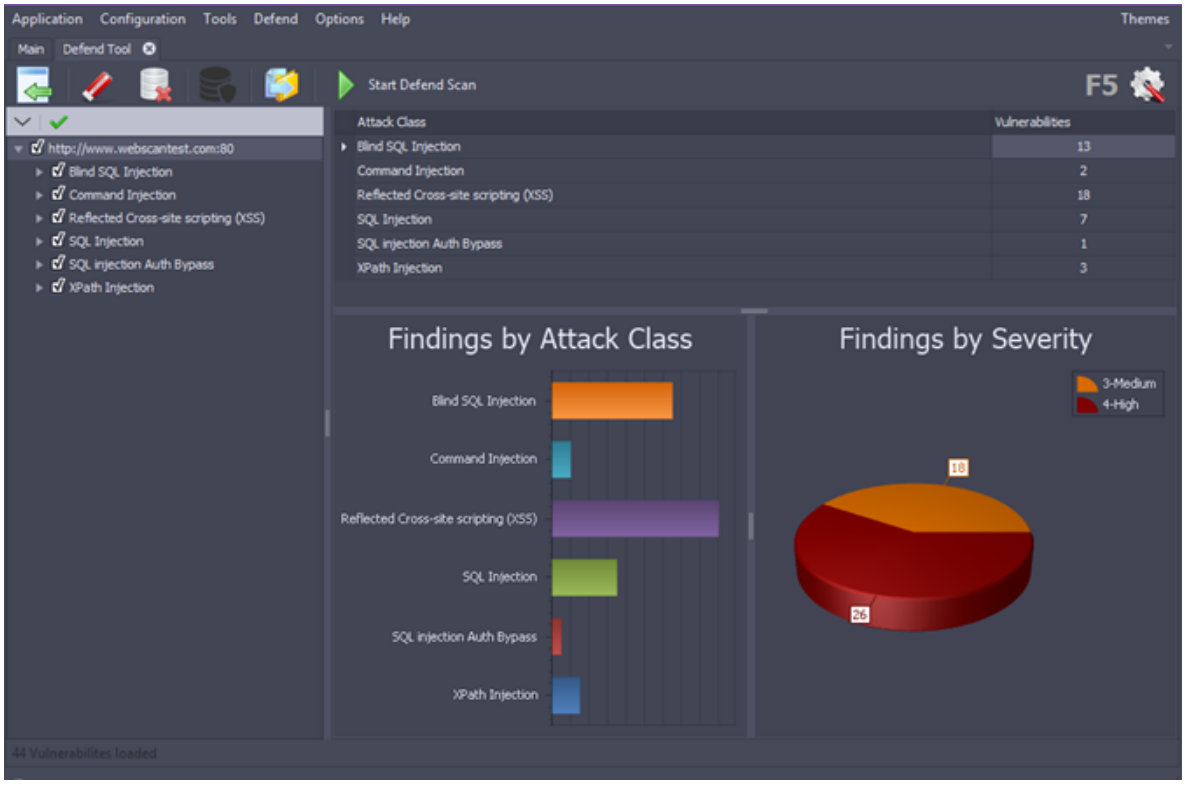
Load Vulnerabilities

Import your desired VulnerabilitySummary generated in AppSpider

- Click on the Load Findings icon.
- The open VulnarabilitySummary widow will open to allow you to select the vulnerability summary XML file for the desired AppSpider scan that has already been completed. The default location for the scan results files is Documents – APPSpider- Scans. From here you can click on the desired scan results directory and navigate to the Reports folder to find the vulnerability summary XML file.

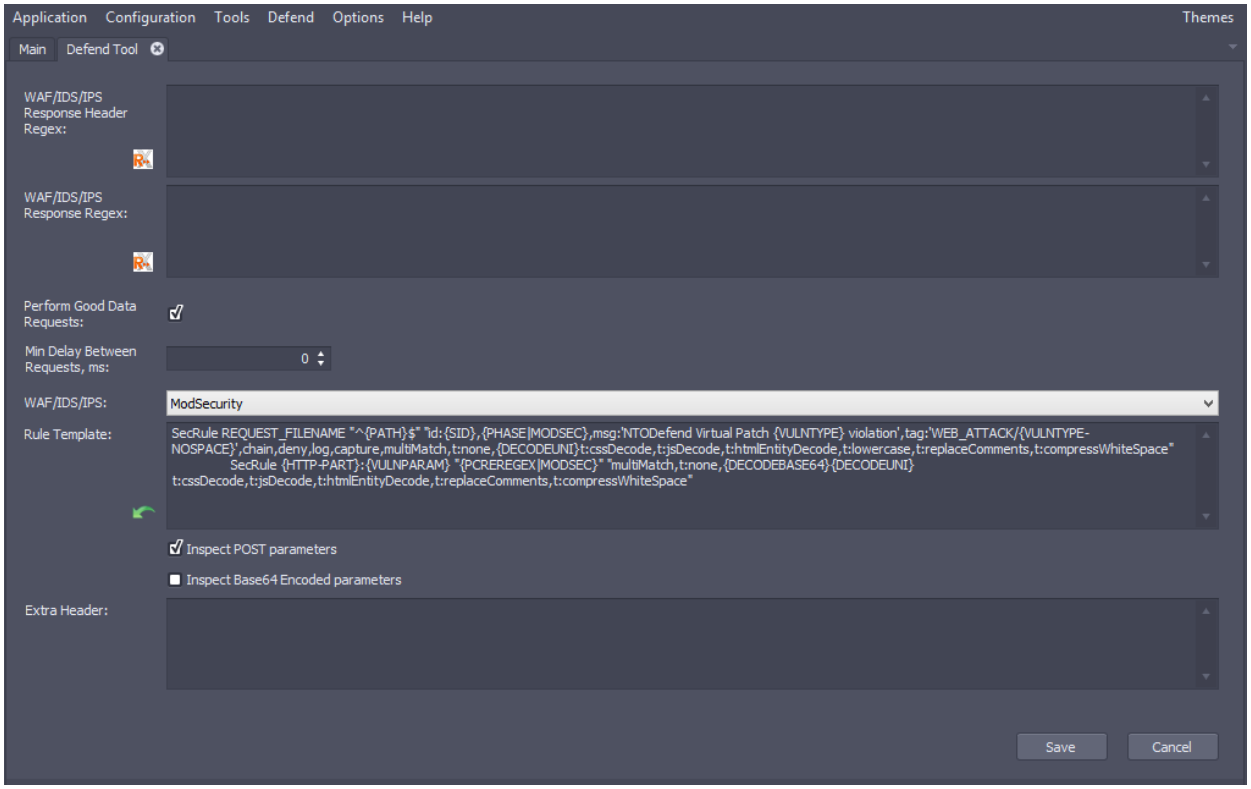


Once the vulnerability summary XML file is selected and the open file icon is pressed the vulnerability summary data will be loaded into Defend and once complete you should see a screen similar to below.



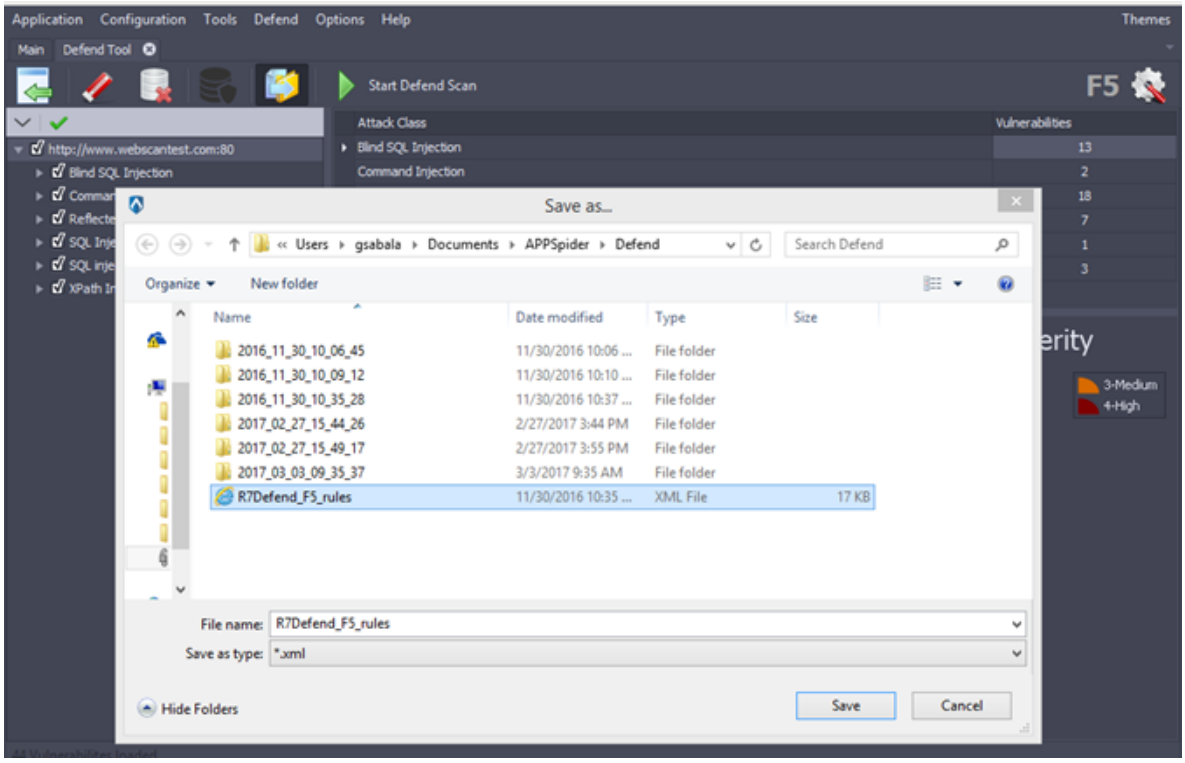
On this UI screen you can select which vulnerabilities Defend should generated WAF rules sets for by selecting or deselecting the check box next to the vulnerability name in the left hand column. From this window you can also select the Options icon in the upper right corner to open the Options UI which will allow you to do a variety of modifications.

Options UI:



Once you are finished with the desired modifications on the Options UI screen press the save button which will take you back to the main UI.

When finished with any desired modification the Defend rules can be exported to a file by clicking on the Export Rules icon. Once the icon is selected a Save as window will appear to allow you to choose the location of the Defend rules file.

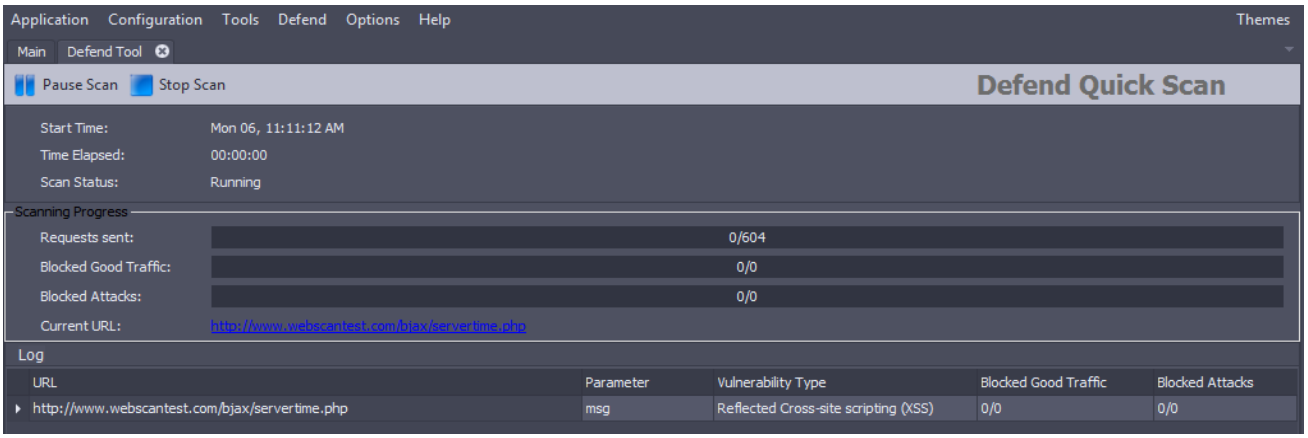


Import Custom Rules

The Defend rules file which was created can now be loaded into your WAF/IDS/IPS by following the custom rules import process defined by your application vendor.

Defend Scan

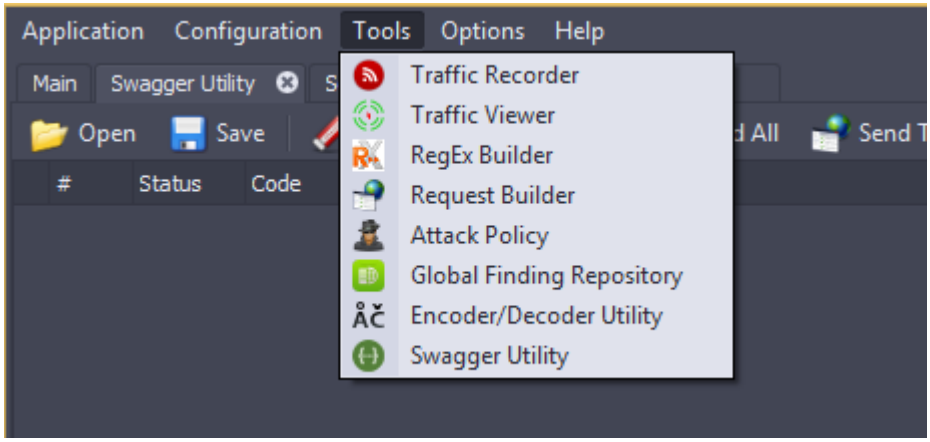
Once the Defend rules have been successfully uploaded into your WAF/IDS/IPS solution you can initiate a Defend scan which will replay the attacks AppSpider used to discover the vulnerabilities to confirm that they are no longer exploitable.



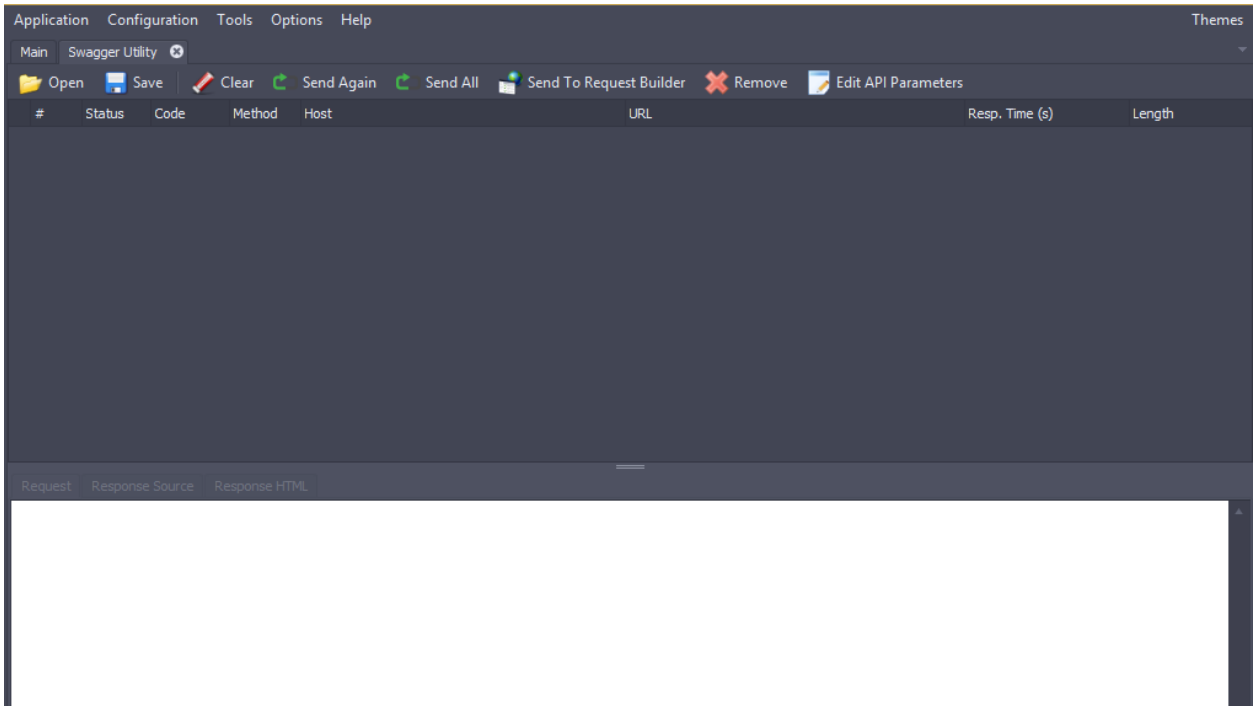
Swagger Utility Introduction:

AppSpider Pro is now capable of testing Swagger-enabled API's which further automates the process of testing APIs within AppSpider by eliminating the need to capture traffic via a proxy prior to testing. Now, you can simply upload a Swagger file to AppSpider and then AppSpider leverages Universal Translator to analyze the file and then discover vulnerabilities in the API. This should prove to be a significant time savings and enable your team to test more of your APIs than before. Swagger, an open source solution, is one of the most popular API frameworks. It defines a standard interface to REST APIs that is agnostic to the programming language. A Swagger-enabled API enables both humans and computers to discover and understand the capabilities of the service. AppSpider parses the swagger document to generate function calls and create values for the expected parameters. The file is then saved as a TREC traffic recording file which then can be used by AppSpider to scan and attack the REST API. The Swagger Utility currently supports the Swagger 2.0 version saved in JSON.

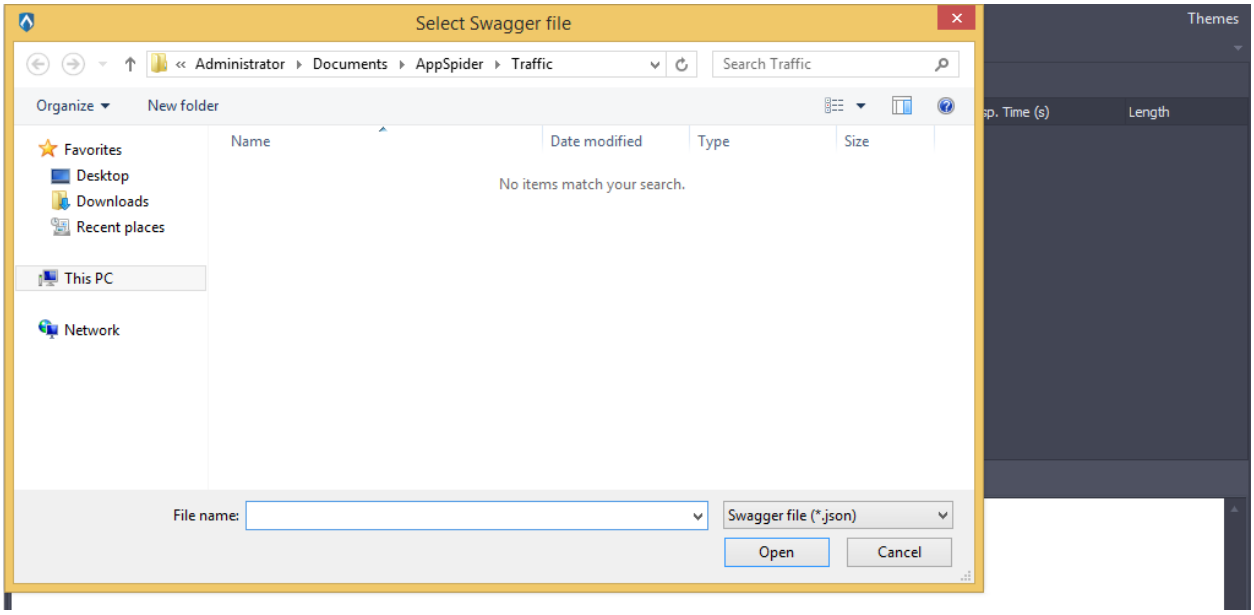
The Swagger capability is accessible within the Tools section in AppSpider.



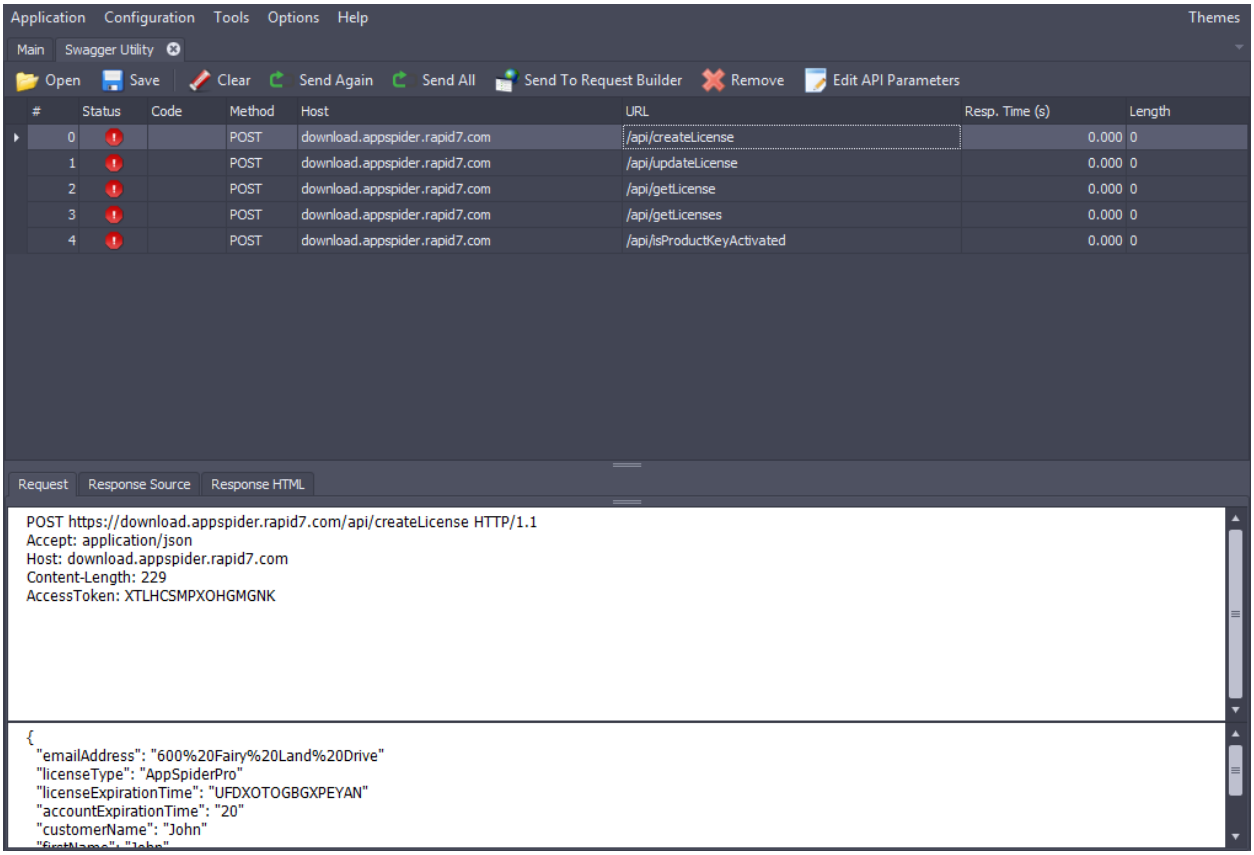
Once the “Swagger Utility” icon is clicked on a new UI window (see below) is opened with the tab titled “Swagger Utility”



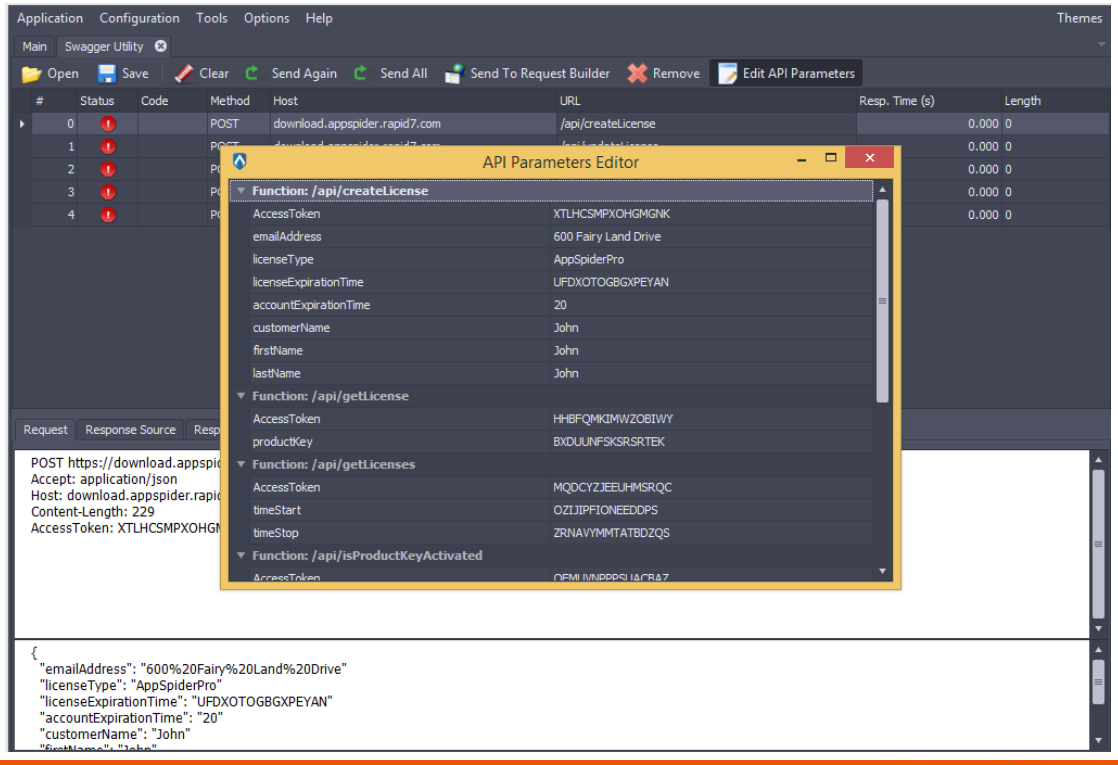
Here you can click on the “open” icon which will open a file selection dialog box (see below) to allow the selection of which Swagger JSON files should be uploaded to AppSpider for scanning.



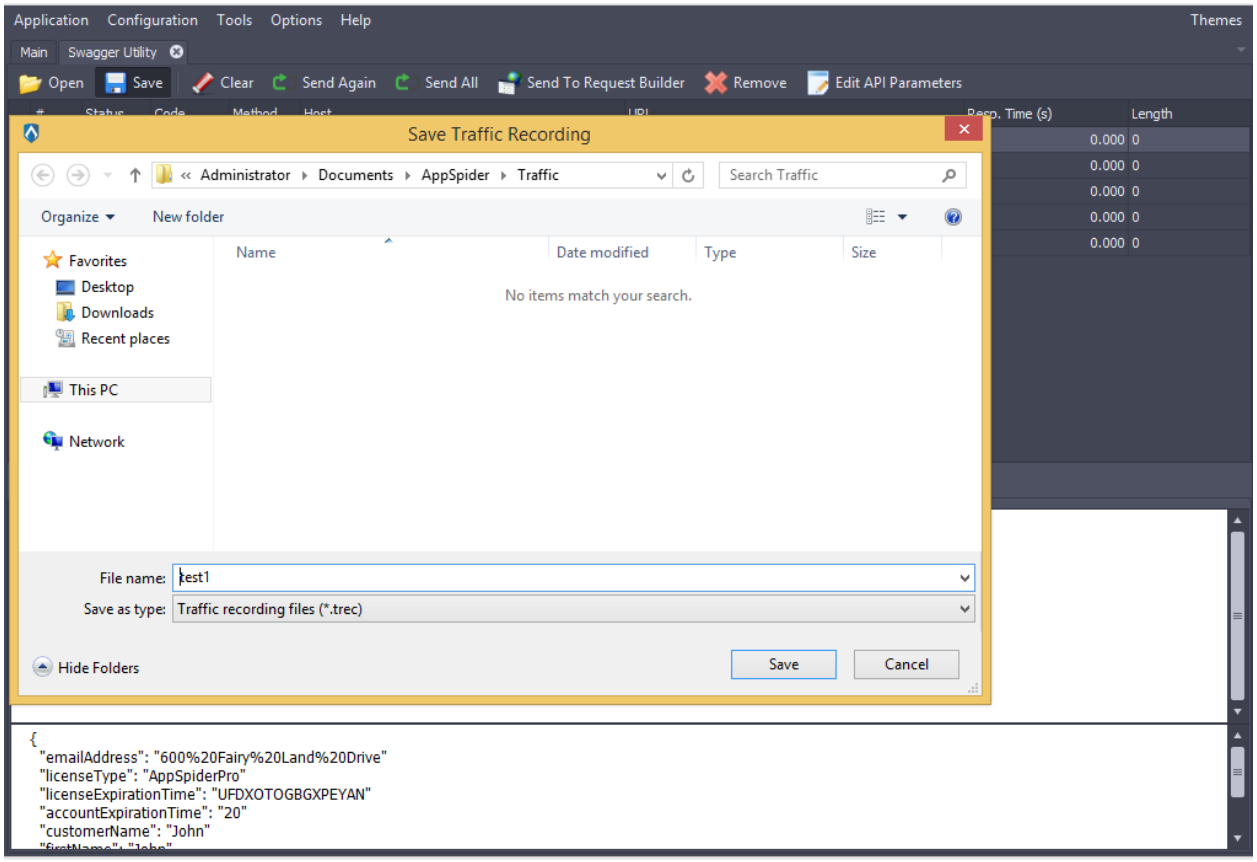
Once the Swagger JSON file has been selected and the open button is pressed the various API function calls are listed in the traffic viewer window (see below).



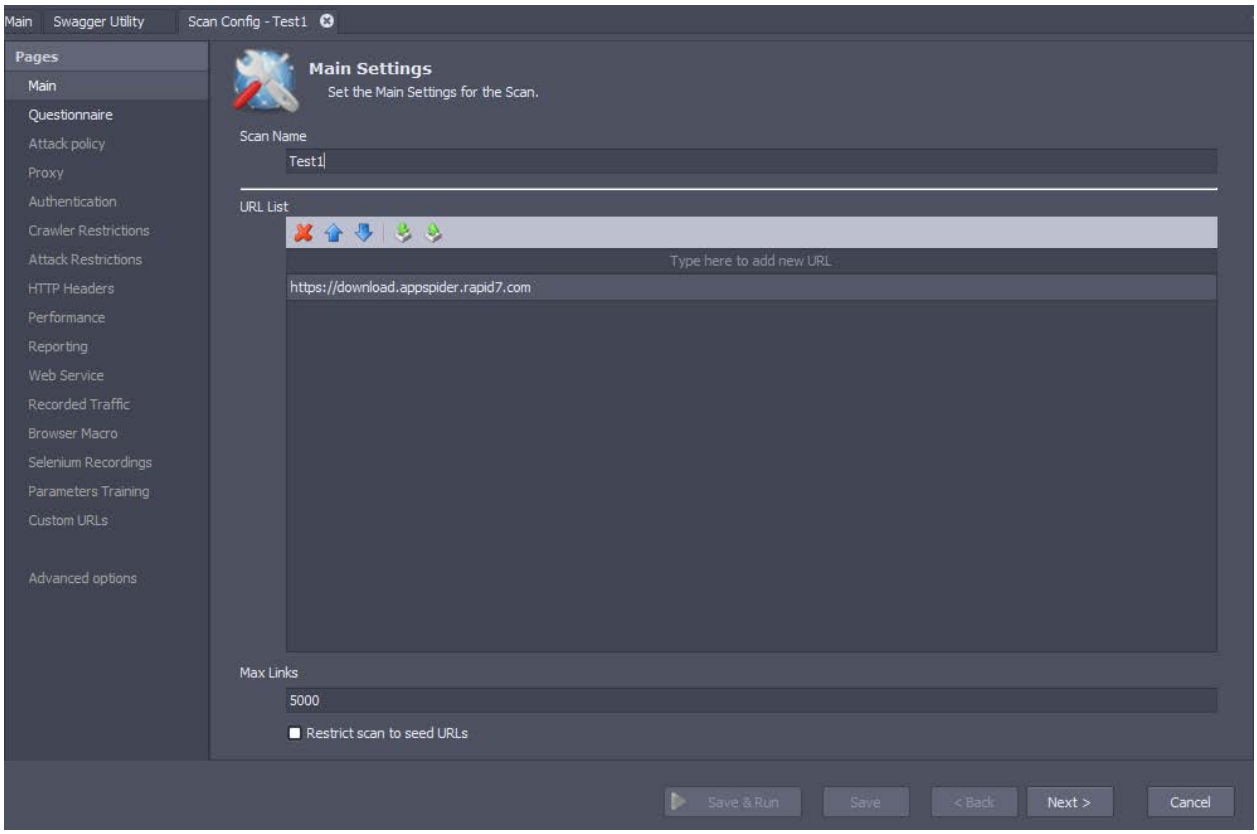
The “Edit API Parameters” button opens a UI (see below) which allows the editing of the various parameters. Once the needed edits are made simply close out the UI box and the updated parameter values will be propagated in the various request calls.



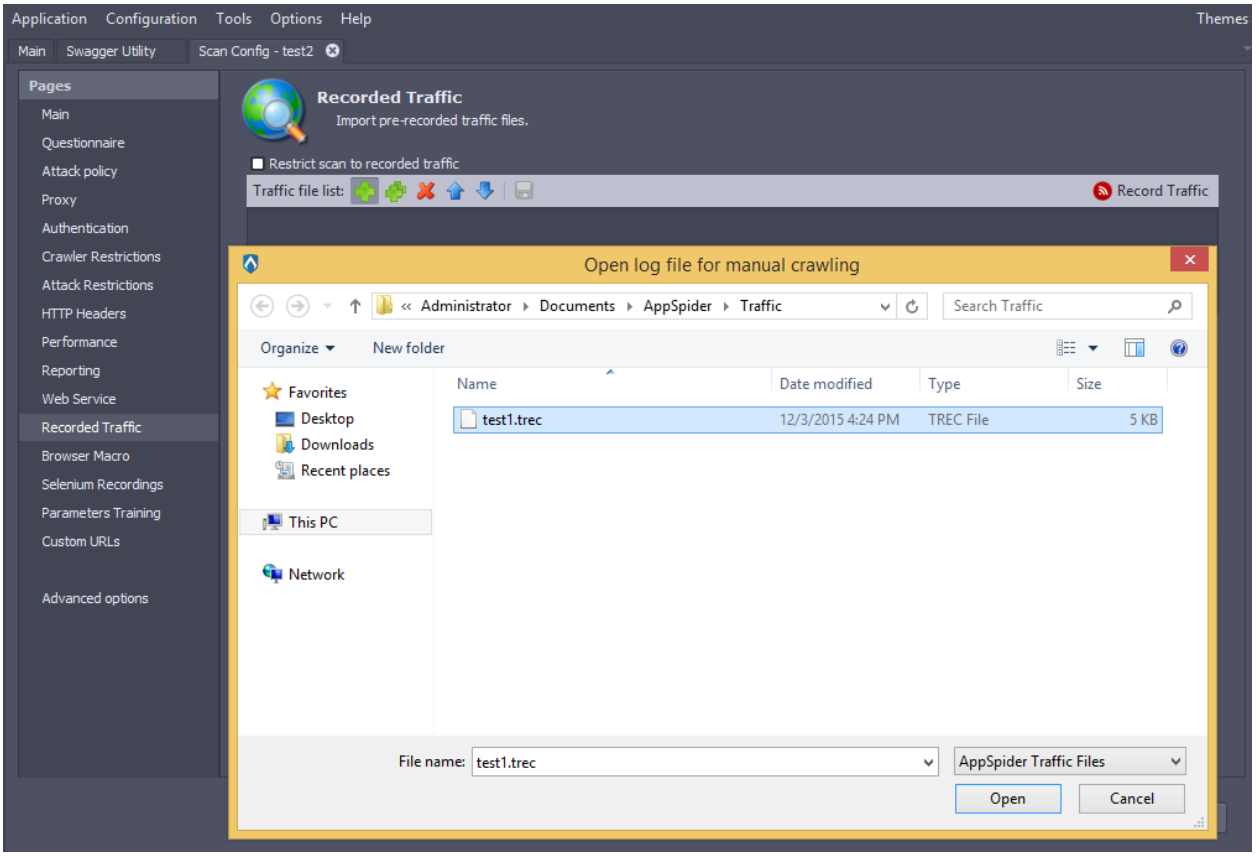
Upon completing any needed editing click on the save button to save the traffic recording file which can be added to a scan configuration for scanning by AppSpider.



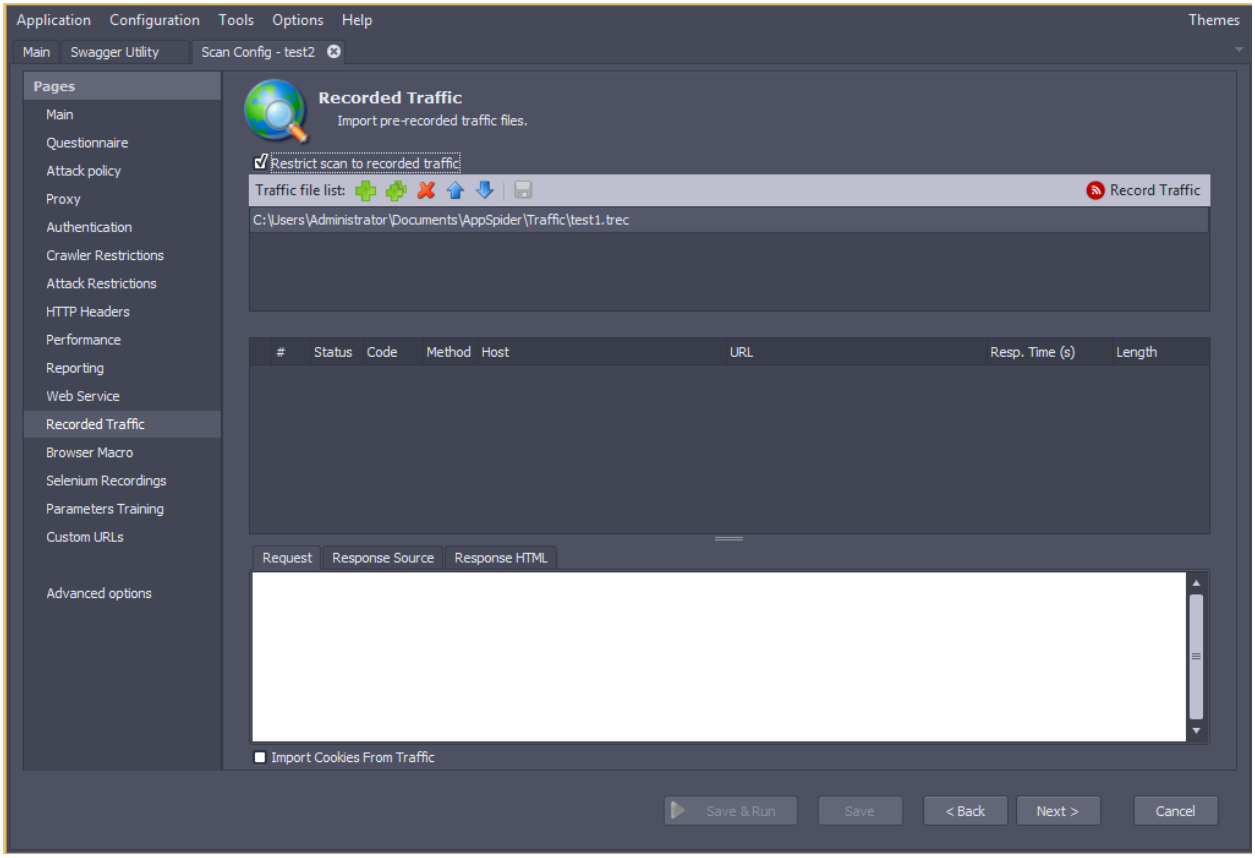
Now you can create a new scan configuration where the traffic recording file that was created by the Swagger Utility can be loaded for scanning. When creating the scan configuration it is recommended that you include the base URL of the REST API in the URL list so that the same domain restrictions that apply to that will apply to the REST calls.



Once you get to the Recorded Traffic section of the scan configuration press on the Green Plus icon (+) to add the Traffic recording file created by the Swagger Utility.



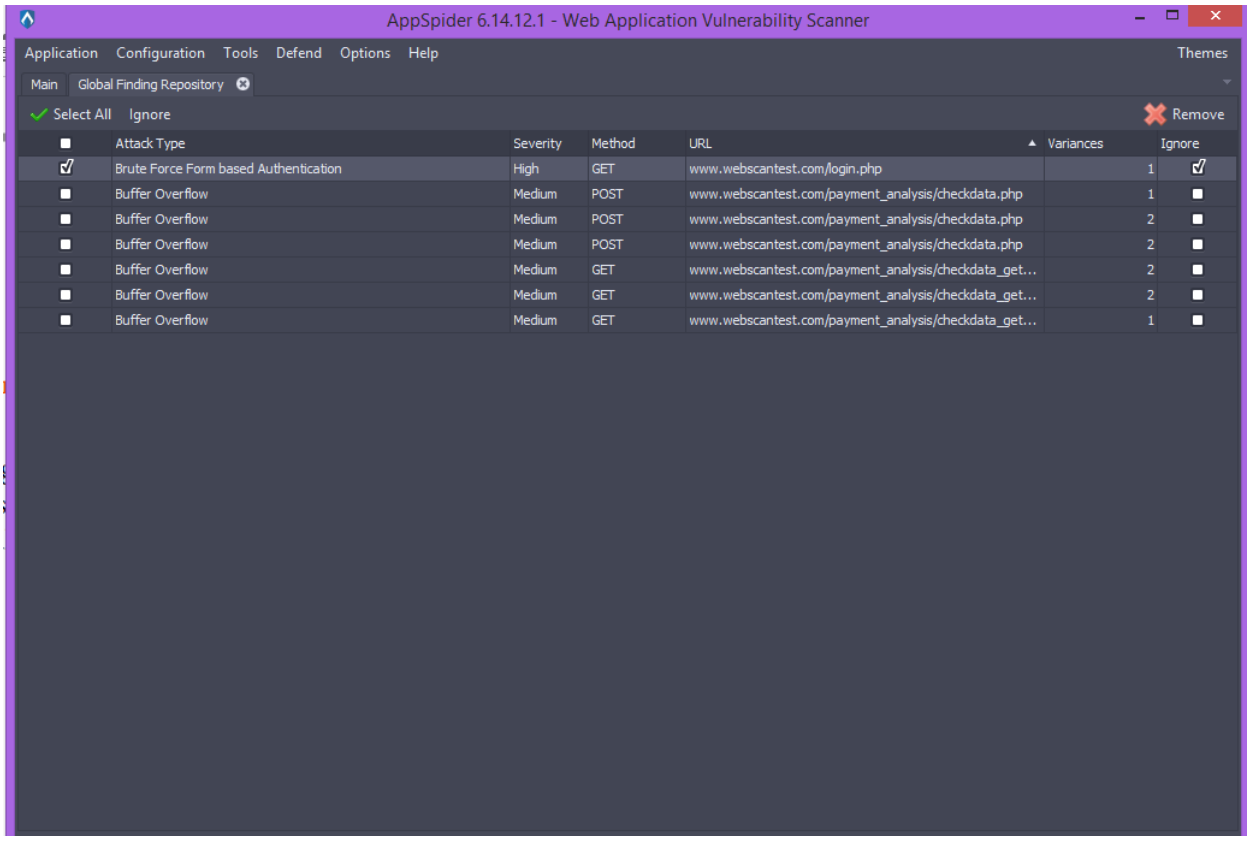
If you would like to limit the scan to only the recorded traffic file then check the restrict scan box towards the top of the UI.

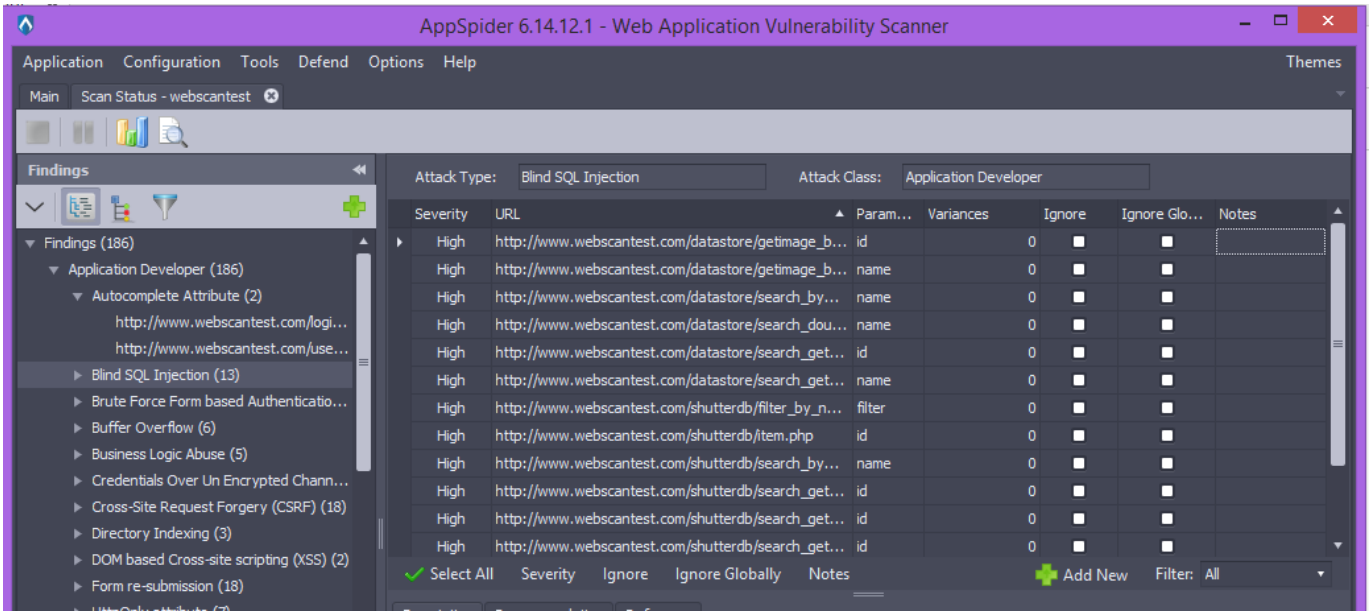


Once the remaining scan configuration is completed then click on the Save & Run button to kick off the scan.

Global Repository

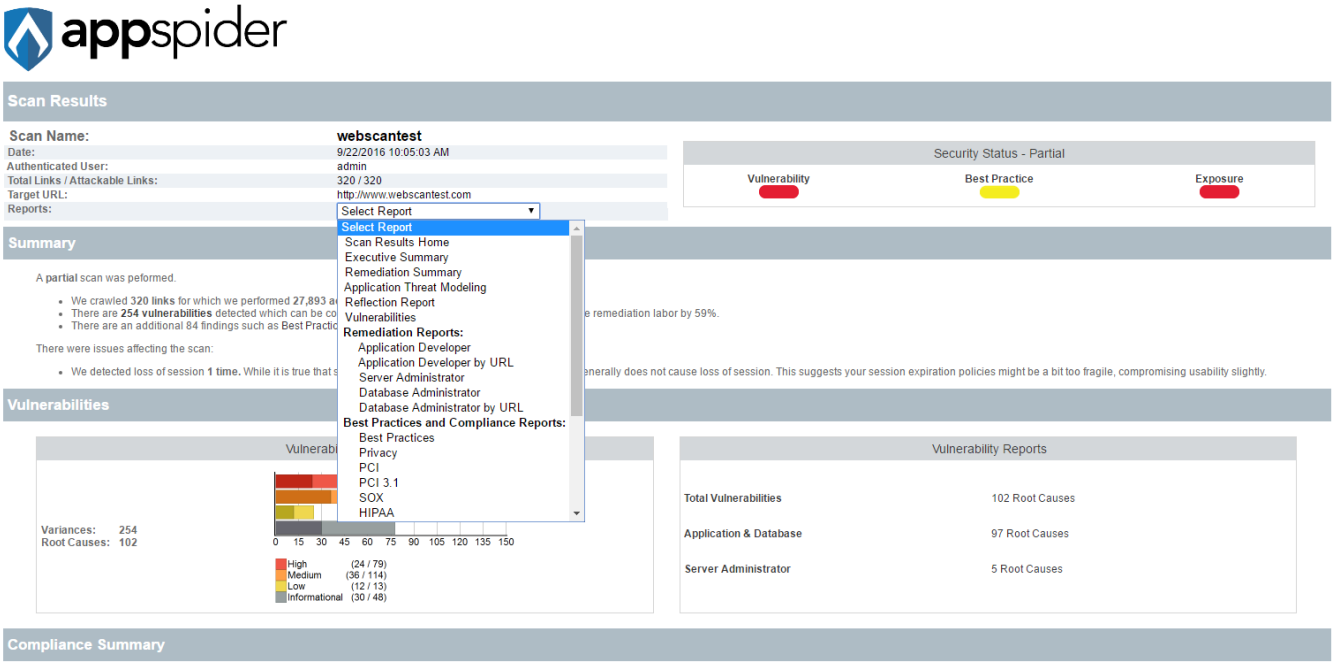
Findings can be declared a False positive by user. This can be done either within a scope of a single scan, or globally. When finding is ignored globally two things happen: the finding gets into a global repository, and it will be excluded from all the results & reports generated on that system.



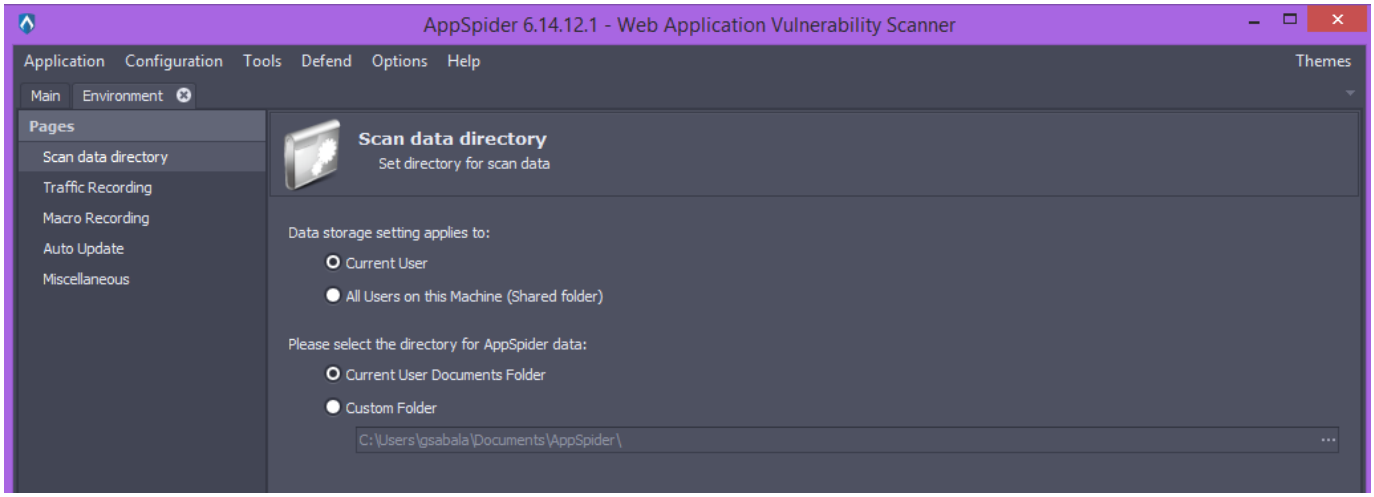


HTML Reports

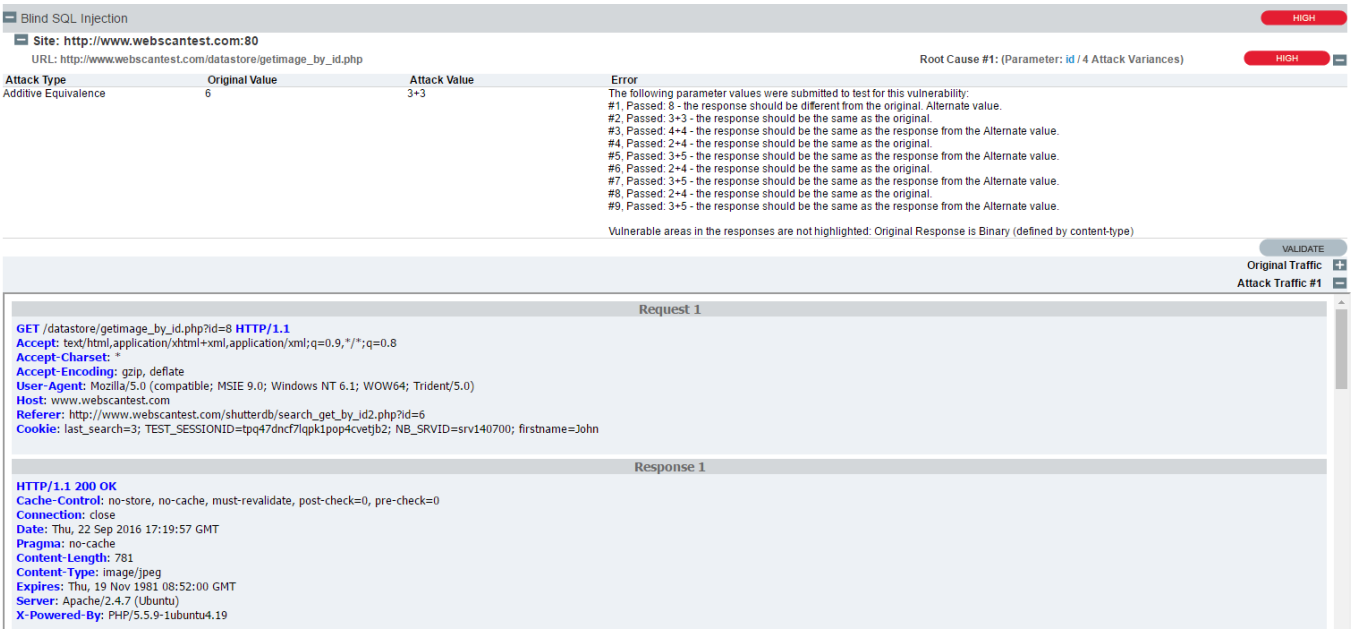
Once a AppSpider scan is completed a HTML report is automatically generated and a browser window should be opened showing the HTML report. The HTML report can also be accessed by clicking on the HTML report icon  on either the scan console or scan status screens.



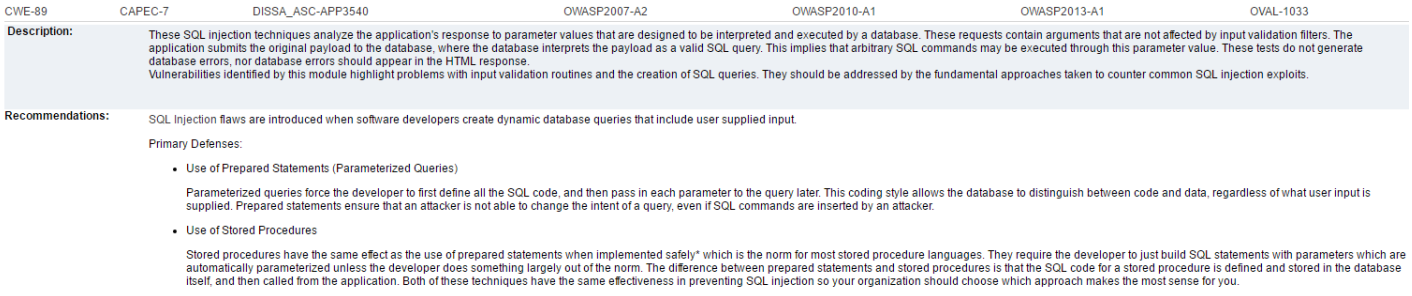
When the HTML report is opened it should appear like the screen shot above. By clicking on the select report drop down you will see several report options available which will present the AppSpider scan results in different formats depending on the theme of the report selected. The HTML report files can be located in the report folder of the scan located in the AppSpider scan data directory. The location of the scan data directory can be found by clicking on the Options menu and then clicking on the environment tab to get to the UI screen below.



The various findings in the report will have a variety of data what will assist in prioritizing how your development team should addressed the found issues. All findings will have a severity rating and information on where the finding is located in the application and the request/response traffic to assist in remediation.

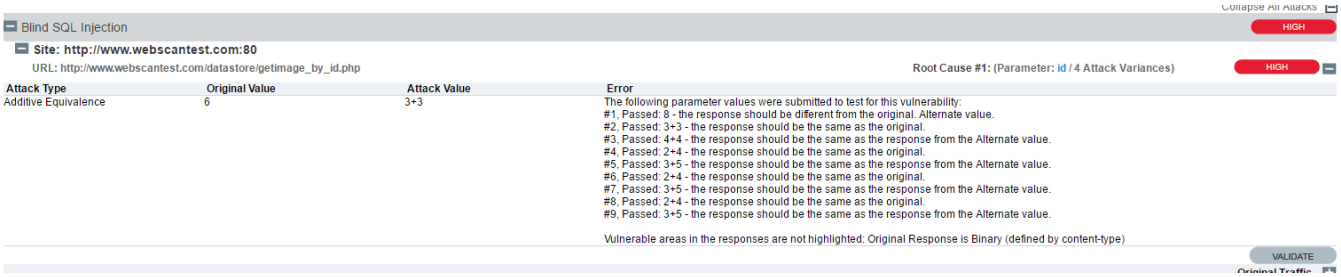


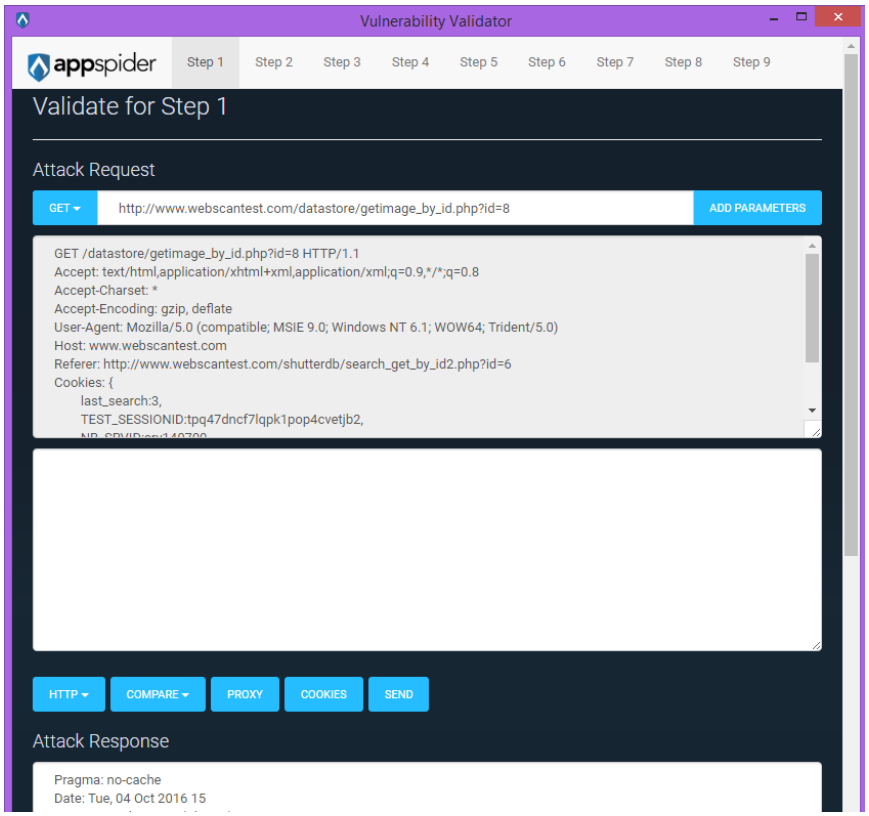
Each finding will also have a description of what the vulnerability is and recommendations on how the issue can be resolved. There are also links to third party application security organizations documentation on the vulnerability such as OWASP that will assist in helping to understand the nature of the vulnerability detected.



Validate App

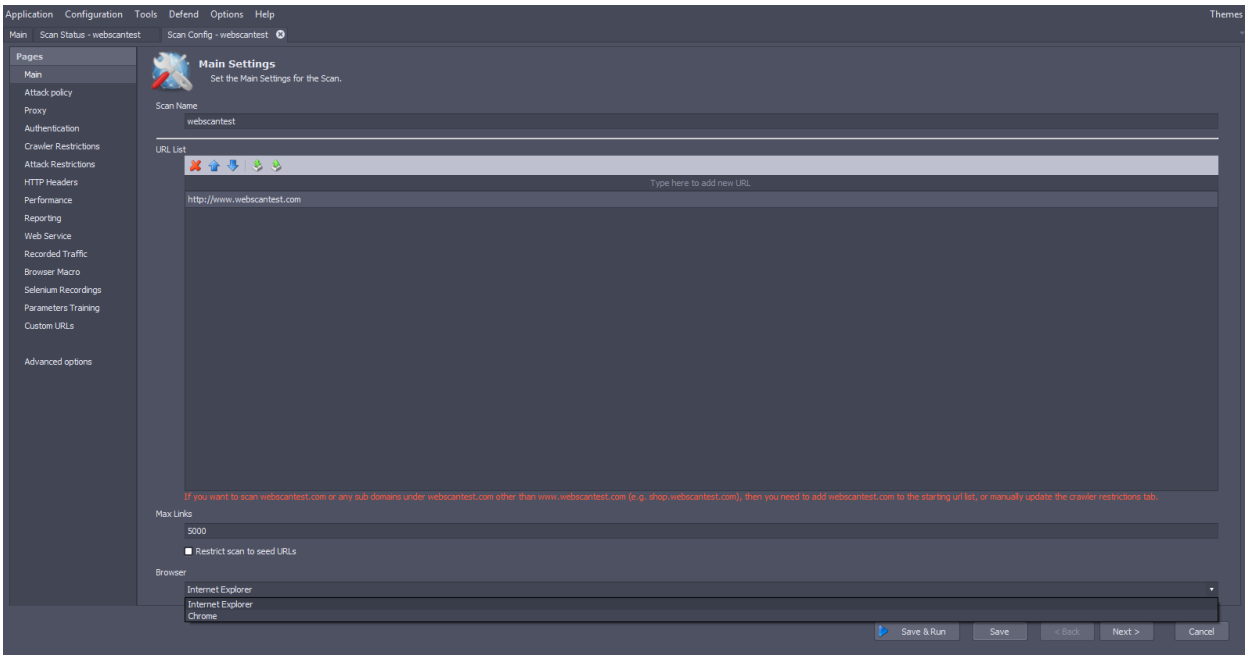
Within each finding you will see a "Validate" button which when pressed will launch a java based applet which will replay the attack and see the response which identified the vulnerability in the application. If you are using the Google Chrome browser then you will need to install the AppSpider browser plug in which can be found in the Chrome extensions store or by clicking on the following URL: https://chrome.google.com/webstore/detail/appspider-validate-chrome/decadhidkojjhcbbjhljnofgejilmmkl?hl=en-US.



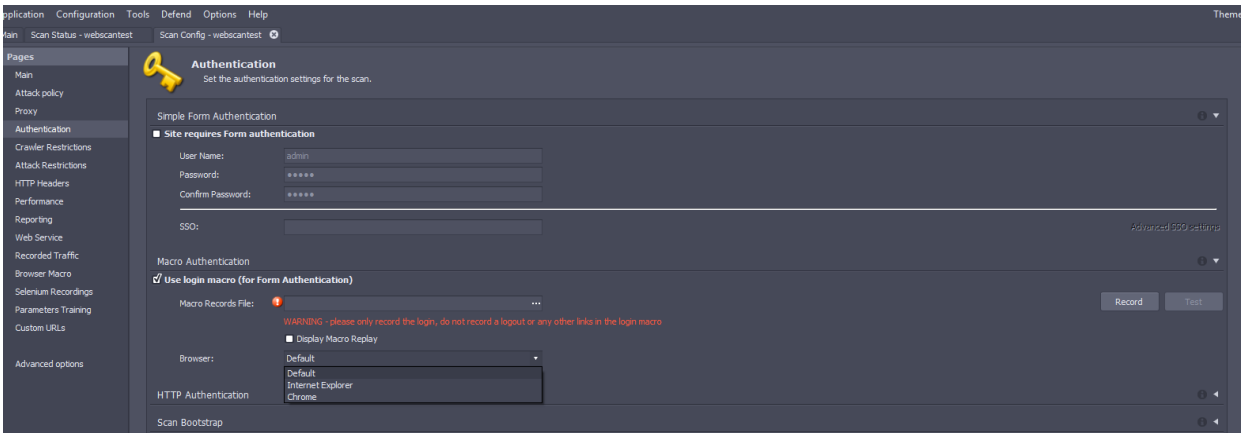


Chrome/WebKit Integration:

The Chrome/WebKit browser has been added as a configuration option along with the default browser option of Internet Explorer within AppSpider Pro. AppSpider Pro relies on the integrated browsers to facilitate the crawling and attack functionality. In some uses cases a web application will function differently depending on what Browser is being used, and thus Chrome has been added as an option to provide additional coverage options. The Chrome browser can be selected in two ways, first on the Main Settings page of the scan configuration set up towards the bottom of the page you will see a selection box entitled browser which you can select Chrome.



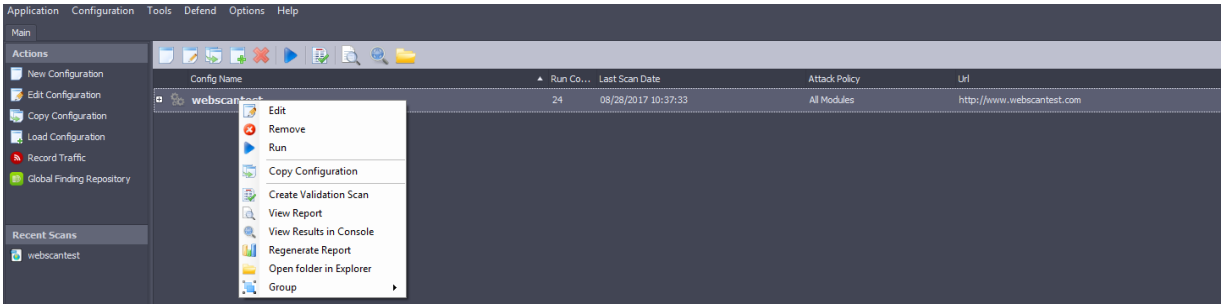
The second option is within the Authentication page of the scan configuration if the Macro Authentication is enabled you will see a Browser selection box where Chrome can be selected.



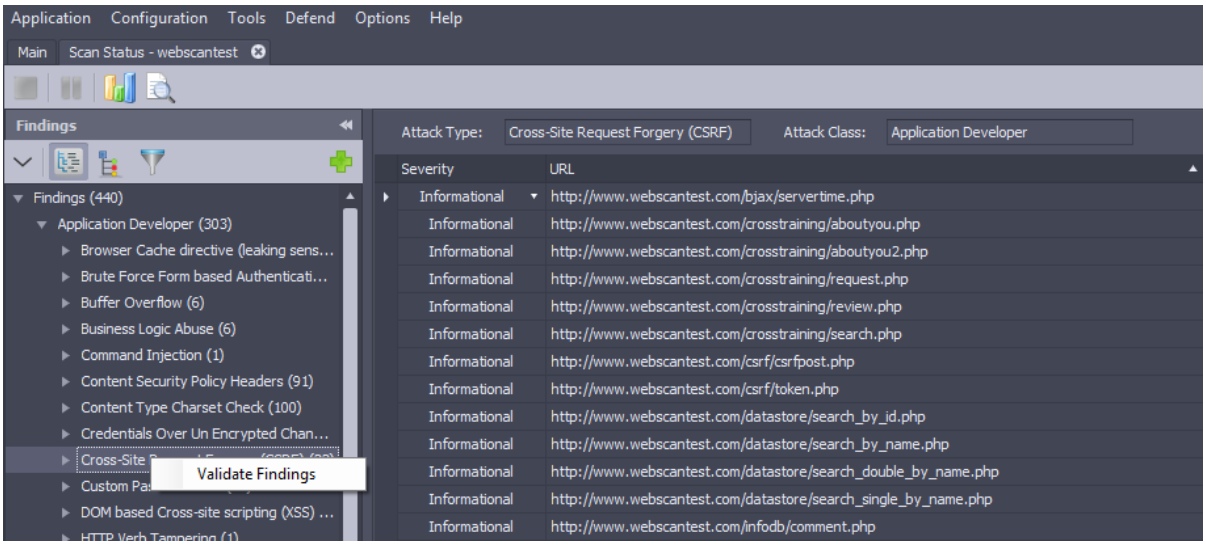
Once the Chrome Browser is selected the scan configuration can be completed and then the scan can be run.

Validation Scan

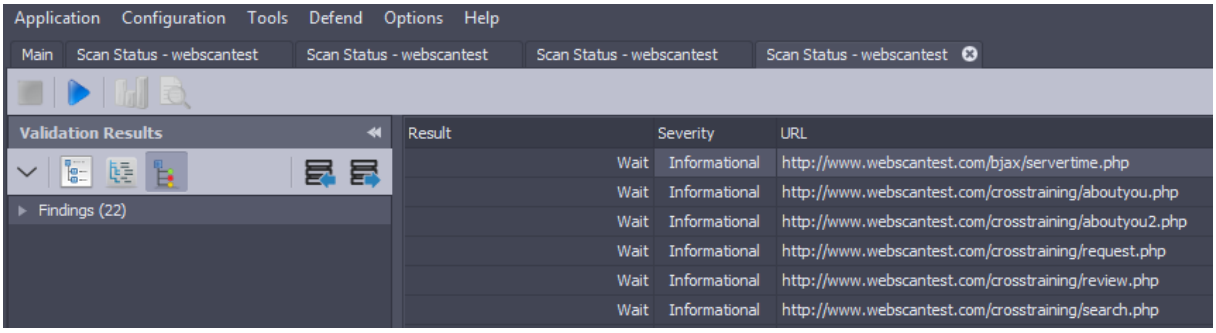
A new scan method has been added to AppSpider Pro which will allow user to run a AppSpider scan against a previously scanned target to verify that a vulnerability or set of vulnerabilities has been resolved after a fix in the code has been deployed by your development team. The validation scan will be a quicker way of validating resolutions to discovered vulnerabilities than re-running a complete scan of a web application. Users can kick off a validation scan in three ways, first by placing the mouse cursor over the desired scan configuration in the main screen and doing a right click to select ”create Validation Scan”.



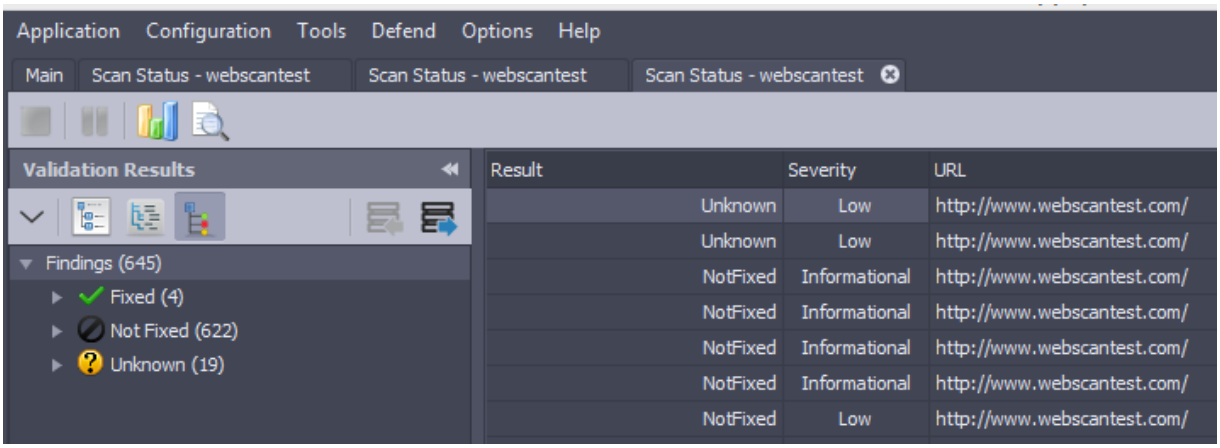
Second option is to highlight the desired scan configuration in the main UI window and then click on the Create Validation Scan icon located above the list of available scan configurations. The third/final option to kick off a validation scan is to load the results of a previous scan into the AppSpider Pro Console and then using your cursor highlight the vulnerability you wish to Validate and then right click and click on the Validate Findings popup to kick off the validation scan.



Second option is to highlight the desired scan configuration in the main UI window and then click on the Create Validation Scan icon located above the list of available scan configurations. The third/final option to kick off a validation scan is to load the results of a previous scan into the AppSpider Pro Console and then using your cursor highlight the vulnerability you wish to Validate and then right click and click on the Validate Findings popup to kick off the validation scan.



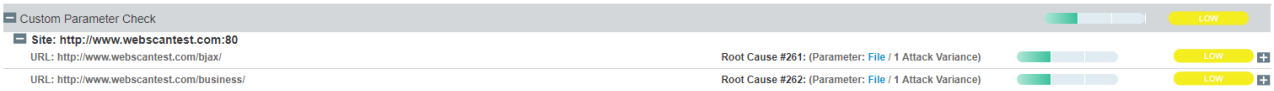
Once the Validation scan has been completed results will be visible within the results UI screen below and in HTML report. Fixed indicated that the vulnerability has been resolved, Not Fixed indicates AppSpider was able to discover the vulnerability. In some instances AppSpider may not have enough information to confirm if a previously discovered vulnerability has been fixed and in these instances it will be listed as Unknown.



Confidence level for findings

A Confidence level for findings in HTML and JASON reports has been added to provide users with a visual indicator of how certain AppSpider is that a particular finding is valid. AppSpider uses the response data it receives during the scanning process to visually display a high (3 bars highlighted), medium (2 bars highlighted), or low (one bar highlighted).

Low Confidence indication



Medium Confidence indication

High Confidence indication