Burp Suite Extensions 1.0

In this section:

Introduction

The Parasoft Burp Suite Extensions package enables you to perform security and penetration testing against APIs and browser-based web applications using SOAtest test scenarios with the Burp Suite web application security assessment tool. The package contains a "Burp Extender" that is installed into Burp Suite, as well as two tools that are used within SOAtest:

- Burp Suite Analysis Tool: Sends data to Burp Suite for security analysis and reports Burp Suite findings within the SOAtest user interface.
- Burp Suite Reporter: Generates a Burp Suite report capturing security analysis findings.

Requirements

- SOAtest 9.10 or later
- A SOAtest .tst with at least one Browser Playback tool or test client (e.g., SOAP, REST, EDI, or Messaging Client)
- Burp Suite Professional 1.7.03 or later

Installing the SOAtest Tools

The SOAtest tools are packaged into the soatestburpsuitetools.jar file, which can be installed from the UI or command line.

UI Installation

1. Choose **Parasoft> Preferences**.
2. In the System Properties preferences page, click **Add JARs**.
3. Browse to and choose the soatestburpsuitetools.jar file to add it to the SOAtest classpath.

Command Line Installation

Add the soatestburpsuitetools.jar file to the system.properties.classpath property in your localsettings properties file.

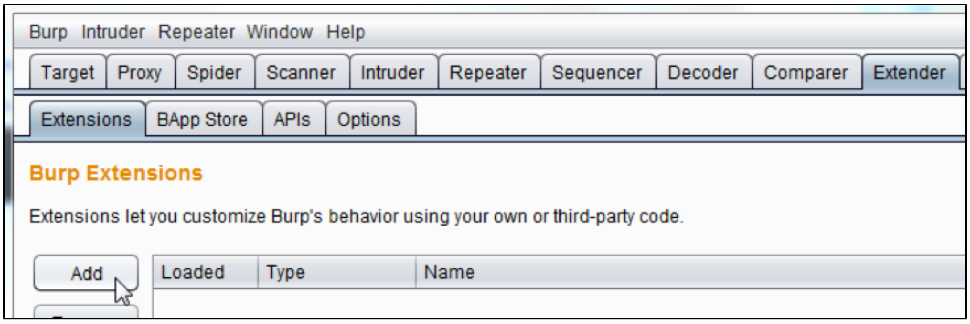
For example:

```
system.properties.classpath=<path to jar>/soatestburpsuitetools.jar
```

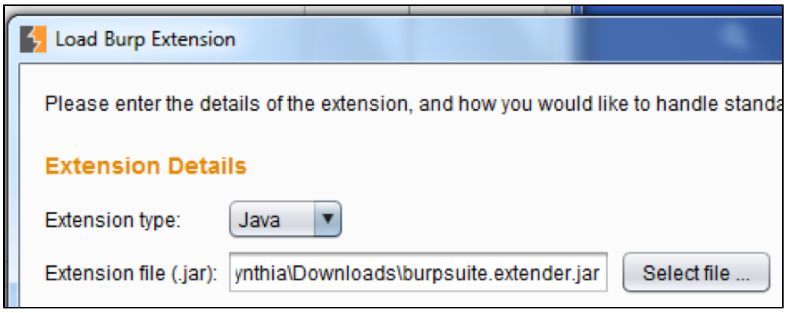
Installing the Burp Extender

Install the Parasoft SOAtest Burp Extender (burpsuiteextender.jar) following the standard Burp Extenders installation process.

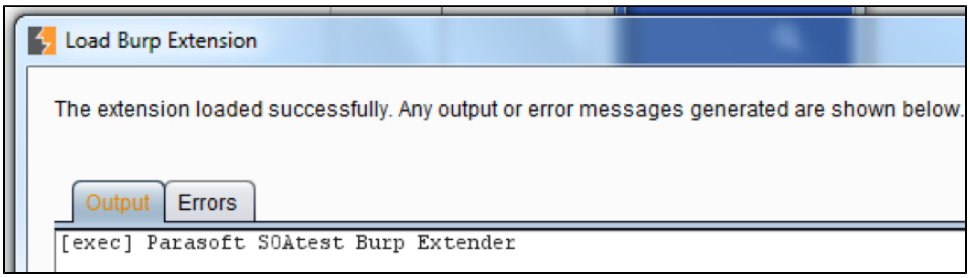
1. Launch the Burp Suite UI.
2. In the Extender> Extensions tab, click **Add**.



3. Under Extension Details, ensure that Extension type is set to Java, then browse to the burpextender.jar file for the Extension file (.jar) field.



4. Click **Next**. You should see a message that says “Parasoft SOAtest Burp Extender” in the Output tab.

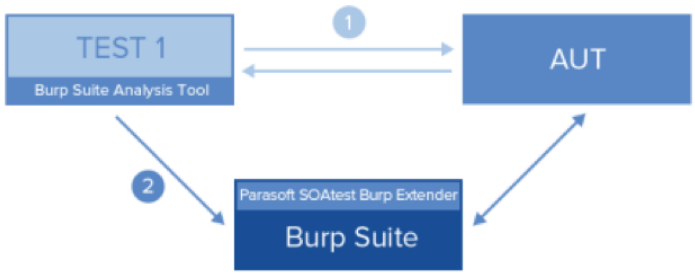


5. Verify that no errors are reported in the Errors tab, then click **Close**.

Usage Overview

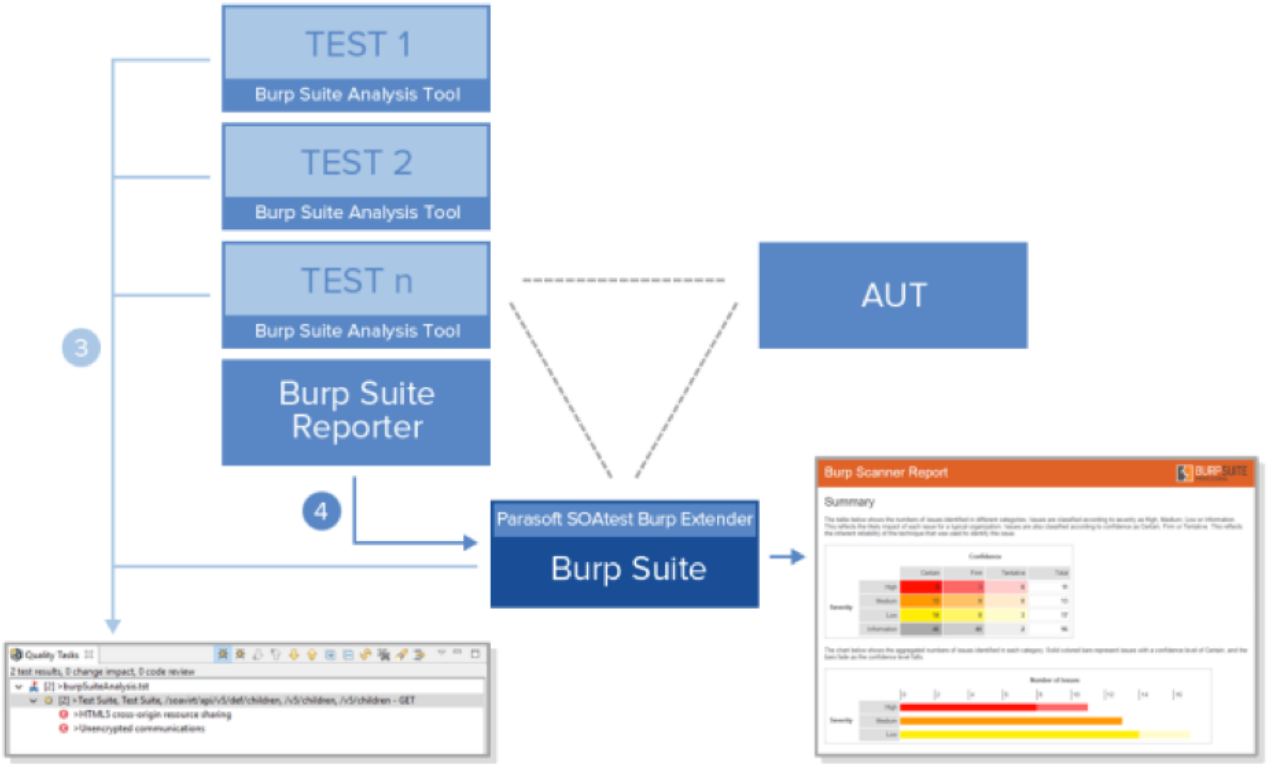
On the Burp Suite side, the Parasoft SOAtest Burp Extender extends Burp Suite with a simple HTTP server that enables Burp Suite to communicate with the SOAtest Burp Suite Analysis tools. On the SOAtest side, Burp Suite Analysis Tools are added throughout each SOAtest API/browser test scenario that you want to test with Burp Suite.

Upon execution, each test with an attached Burp Suite Analysis Tool interacts with the AUT as normal (see #1 below), then sends data to Burp Suite for security analysis (see #2 below). Burp Suite interacts with the AUT to perform its analysis.



As each test is executed, results are sent to SOAtest's Quality Tasks view (see #3 below). These results can be saved as an XML report, then uploaded to Parasoft DTP (see Connecting to Parasoft Development Testing Platform).

At any point where a Burp Suite Reporter tool is executed, a Burp Suite HTML report with advanced security details is generated (see #4 below).



Because Burp Suite analysis can impact the behavior of functional test scenarios and takes much longer to run than typical test scenarios, we strongly recommend that you maintain two difference copies of your test scenarios: one that is used for your functional test runs, and another that is used for your security test runs. Whenever you want to perform security tests on your application, make a copy of the latest version of your functional test scenarios, then add Burp Suite tools to the copy. This way, your original tests can still be used for functional testing—without any behavior or performance impact.

The general workflow for enabling this is:

1. Identify the test scenarios that you want to use for security testing and copy them. You can continue executing the original test scenarios for functional testing as normal.
2. Add the Burp Suite tools to the copied test scenarios, then execute them as security tests.
3. As the application changes, update only the functional test scenarios. Whenever you are ready to run the corresponding security test scenarios, repeat the above process of copying from the latest set of functional tests and then configuring the copy for security testing.

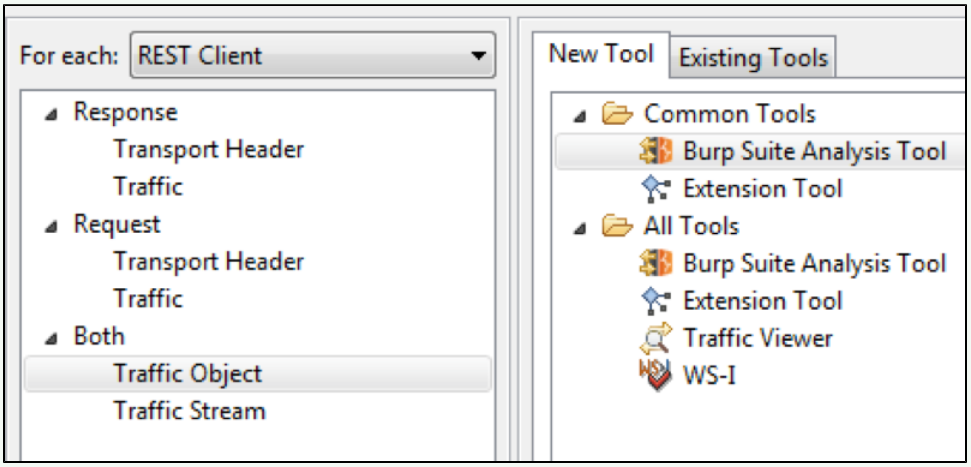
Configuring a Test Scenario for Burp Suite Testing

1. Attach a Burp Suite Analysis tool attached as an output tool to every Browser Playback tool or test client (e.g., SOAP Client, REST Client, EDI Client, or Messaging Client) in the test scenario. For test clients, be sure to attach it as a Traffic Object output (see Adding Test Outputs).

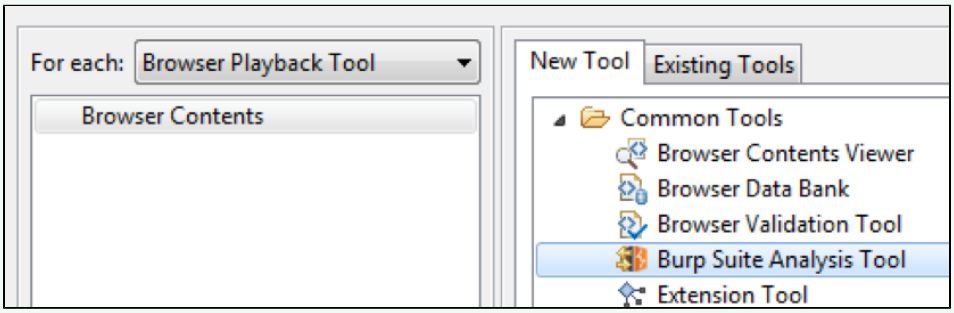
Adding Multiple Outputs

The fastest way to add outputs to all tools in a test scenario is to right-click the top node of the test scenario, choose Add Multiple Outputs, then select one of the following:

- For test clients, select **Both> Traffic Object** on the left, then **Burp Suite Analysis Tool** on the right.



- For Browser Playback tools, select **Browser Contents** on the left (for test clients), then **Burp Suite Analysis Tool** on the right.



2. Add one Burp Suite Reporter tool as a regular (not output) tool at any point in the scenario where you want a report generated. When this tool is executed, a Burp Suite report will be created to include the security analysis results that were reported since either the start of execution (if this was the first Burp Suite Reporter tool) or since the previous Burp Suite Reporter tool was executed.

   For example, assume you have a .tst with 3 test scenarios. If you want 1 report with the security analysis results for all 3 test scenarios, you would add 1 Burp Suite Reporter tool at the end of the third test scenario. However, if you wanted a separate report for each test scenario, you would add 1 Burp Suite Reporter tool at the end of each test scenario—a total of 3 Burp Suite Reporter tools.

   If you want to generate a single report after running all the .tsts in a given project or directory, you might want to create a .tst file that contains only the Burp Suite Reporter tool, and name it in such a way that it is the final .tst file that is executed (for example, “zzz_reporting.tst”, since .tst files get run in alphabetical order).

3. Configure the Burp Suite Reporter tool's Report Location option to indicate where the Burp Suite HTML report should be saved. If this field is empty, the report will be saved to [${BurpSuiteWorkingDirectory}/SOAtestBurpExtender/report_YYYY-MM-DD_HH-MM-SS.html]

Executing Tests

You can perform security testing after you've configured a test scenario for Burp Suite.

1. Launch Burp Suite.
2. Enter a URL pattern(s) in the Target> Scope configuration page in Burp Suite to define the Target Scope, otherwise SOAtest analysis requests will be ignored.
3. Execute the SOAtest test scenarios that are configured for Burp Suite testing and reporting.

When test execution completes, the associated Burp Suite report(s) will be saved in the specified report directory.

The SOAtest quality tasks view, and any reports generated from SOAtest, may show more issues than the report generated directly from Burp Suite using the Reporter tool shows. This is because SOAtest shows an unfiltered list of issues found, some of which may be duplicates. Burp Suite, however, only shows issues that it considers unique, while combining issues it considers to be duplicates.

Using a Non-default Port for Connecting to Burp Suite

By default, the Burp Suite Analysis Tool expects the Burp Suite extension to expose its simple HTTP server at localhost port 9898—and Burp Suite and SOAtest are expected to be on the same machine. If you're using a different port, you need to reconfigure the host/port in both the Burp Suite extension as well as the Burp Suite Analysis Tool that is run within SOAtest.

Configuring the Burp Suite Extension's Port

Pass the following argument when starting up Burp Suite:

```
java -Dburpsuite.extension.port=<new value> -jar burpsuite_pro_v-1.7.03
```

Configuring the Port Used by SOAtest Burp Suite Analysis Tools

Pass the following argument when starting up SOAtest:

```
soatest.exe -J-Dburpsuite.extension.port=<new value>
```

Adjusting the Timeout Settings

In an automation scenario, you might need to configure the timeout settings to handle cases where Burp Suite takes a long time to complete analysis or becomes unresponsive. This will allow your SOAtest test scenarios to continue even if Burp Suite has not completed its analysis. If you do not configure the timeout, no timeout is set and the Burp Suite Analysis Tool will not finish until it receives all results from Burp Suite.

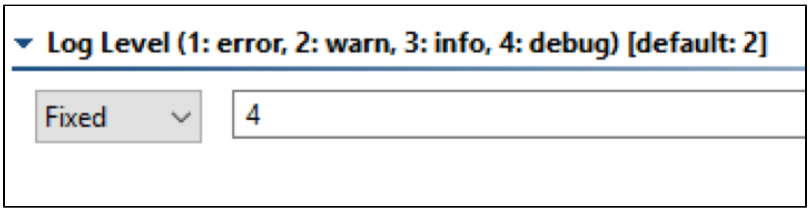
Pass the following argument when starting up SOAtest to configure a timeout:

```
soatest.exe -J-Dburpsuite.extension.timeout=<value>
```

The value is in minutes. A value of `-1` indicates that no timeout should be applied.

Changing the Log Level

By default, the log level for both Burp Suite tools is warn. If you prefer a higher or lower level of details logged to the console and the Event Monitoring view, you can adjust the tool's Log Level setting.



Configuring Severity Levels

You can configure the minimum severity level reported by the Burp Suite analysis tool. By default, the minimum severity level reported is low. As a result, all low, medium, and high violations are reported. You can change the default by passing the following argument when starting SOAtest:

```
-J-Dburpsuite.extension.severity.filter=<severity>
```

The following security levels can be set:

- `high`
- `medium`
- `low`
- `information`

Configuring a higher-level severity means that less severe errors will be ignored. The severity level is not case sensitive.

Third-Party Content

This extension includes items that have been sourced from third parties as outlined below.

- commons-httpclient (Apache license)
- restlet (Apache license)
- jackson (Apache license)

Additional license details are available in this plugin's licenses folder.