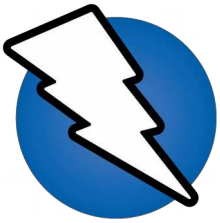
OWASP ZAP 2.7 Getting Started Guide



Overview

This document is intended to serve as a basic introduction for using OWASP's Zed Attack Proxy (ZAP) tool to perform security testing, even if you don't have a background in security testing. To that end, some security testing concepts and terminology is included but this document is not intended to be a comprehensive guide to either ZAP or security testing.

If you are already familiar with security or penetration testing, you may want to start with Introducing ZAP.

See Useful Links for additional resources and information on ZAP.

Security Testing Basics

Software security testing is the process of assessing and testing a system to discover security risks and vulnerabilities of the system and its data. There is no universal terminology but for our purposes, we define assessments as the analysis and discovery of vulnerabilities without attempting to actually exploit those vulnerabilities. We define testing as the discovery and attempted exploitation of vulnerabilities.

Security testing is often broken out, somewhat arbitrarily, according to either the type of vulnerability being tested or the type of testing being done. A common breakout is:

- **Vulnerability Assessment** – The system is scanned and analyzed for security issues.
- **Penetration Testing** – The system undergoes analysis and attack from simulated malicious attackers.
- **Runtime Testing** – The system undergoes analysis and security testing from an end-user.
- **Code Review** – The system code undergoes a detailed review and analysis looking specifically for security vulnerabilities.

Note that risk assessment, which is commonly listed as part of security testing, is not included in this list. That is because a risk assessment is not actually a test but rather the analysis of the perceived severity of different risks (software security, personnel security, hardware security, etc.) and any mitigation steps for those risks.

More About Penetration Testing

Penetration Testing (pentesting) is carried out as if the tester was a malicious external attacker with a goal of breaking into the system and either stealing data or carrying out some sort of denial-of-service attack.

Pentesting has the advantage of being more accurate because it has fewer false positives (results that report a vulnerability that isn't actually present), but can be time-consuming to run.

Pentesting is also used to test defence mechanisms, verify response plans, and confirm security policy adherence.

Automated pentesting is an important part of continuous integration validation. It helps to uncover new vulnerabilities as well as regressions for previous vulnerabilities in an

environment which quickly changes, and for which the development may be highly collaborative and distributed.

The Pentesting Process

Both manual and automated pentesting are used, often in conjunction, to test everything from servers, to networks, to devices, to endpoints. This document focuses on web application or web site pentesting.

Pentesting usually follows these stages:

- **Explore** – The tester attempts to learn about the system being tested. This includes trying to determine what software is in use, what endpoints exist, what patches are installed, etc. It also includes searching the site for hidden content, known vulnerabilities, and other indications of weakness.
- **Attack** – The tester attempts to exploit the known or suspected vulnerabilities to prove they exist.
- **Report** – The tester reports back the results of their testing, including the vulnerabilities, how they exploited them and how difficult the exploits were, and the severity of the exploitation.

Pentesting Goals

The ultimate goal of pentesting is to search for vulnerabilities so that these vulnerabilities can be addressed. It can also verify that a system is not vulnerable to a known class or specific defect; or, in the case of vulnerabilities that have been reported as fixed, verify that the system is no longer vulnerable to that defect.

Introducing ZAP

Zed Attack Proxy (ZAP) is a free, open-source penetration testing tool being maintained under the umbrella of the Open Web Application Security Project (OWASP). ZAP is designed specifically for testing web applications and is both flexible and extensible.

At its core, ZAP is what is known as a “man-in-the-middle proxy.” It stands between the tester’s browser and the web application so that it can intercept and inspect messages sent between browser and web application, modify the contents if needed, and then forward those packets on to the destination. It can be used as a stand-alone application, and as a daemon process.

If there is another network proxy already in use, as in many corporate environments, ZAP can be configured to connect to that proxy.

ZAP provides functionality for a range of skill levels – from developers, to testers new to security testing, to security testing specialists. ZAP has versions for each major OS and Docker, so you are not tied to a single OS. Additional functionality is freely available from a variety of add-ons in the ZAP Marketplace, accessible from within the ZAP client.

Because ZAP is open-source, the source code can be examined to see exactly how the functionality is implemented. Anyone can volunteer to work on ZAP, fix bugs, add features, create pull requests to pull fixes into the project, and author add-ons to support specialized situations.

For more information, see the Zed Attack Proxy Project Page.

As with most open source projects, donations are welcome to help with costs for the projects. You can find a donate button on the owasp.org page for ZAP at https://www.owasp.org/index.php/ZAP.

Install and Configure ZAP

ZAP has installers for Windows, Linux, and Mac OS/X. There are also Docker images available on the download site listed below.

Install ZAP

The first thing to do is install ZAP on the system you intend to perform pentesting on. Download the appropriate installer from ZAP's download location at https://github.com/zaproxy/zaproxy/wiki/Downloads and execute the installer.

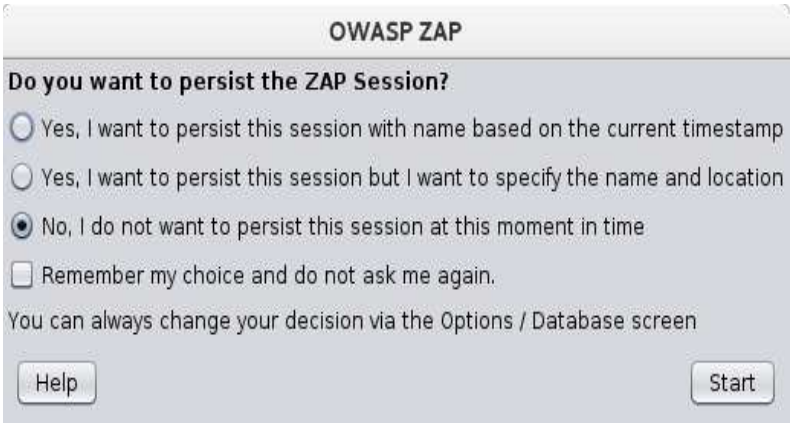
Note that ZAP requires Java 8+ in order to run. The Mac OS/X installer includes an appropriate version of Java but you must install Java 8+ separately for Windows, Linux, and Cross-Platform versions. The Docker versions do not require you to install Java.

Once the installation is complete, launch ZAP and read the license terms. Click **Agree** if you accept the terms, and ZAP will finish installing, then the ZAP will automatically start.

Persisting a Session

When you first start ZAP, you will be asked if you want to persist the ZAP session. By default, ZAP sessions are always recorded to disk in a HSQLDB database with a default name and location. If you do not persist the session, those files are deleted when you exit ZAP.

If you choose to persist a session, the session information will be saved in the local database so you can access it later, and you will be able to provide custom names and locations for saving the files.

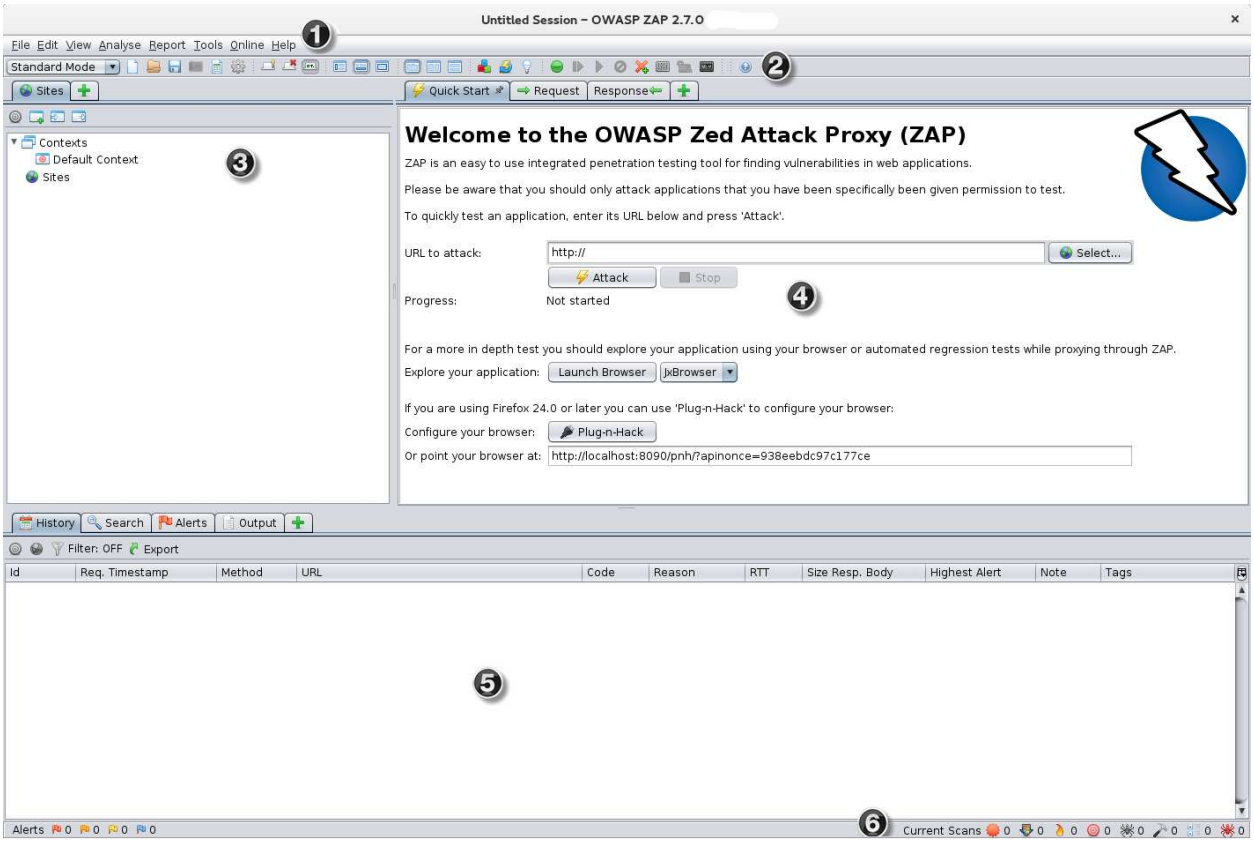


For now, select **No, I do not want to persist this session at this moment in time**, then click **Start**. The ZAP sessions will not be persisted for now.

ZAP UI

The ZAP UI is composed of the following elements:

1. **Menu Bar** – Provides access to many of the automated and manual tools.
2. **Toolbar** – Includes buttons which provide easy access to most commonly used features.
3. **Tree Window** – Displays the Sites tree and the Scripts tree.
4. **Workspace Window** – Displays requests, responses, and scripts and allows you to edit them.
5. **Information Window** – Displays details of the automated and manual tools.
6. **Footer** – Displays a summary of the alerts found and the status of the main automated tools.



While using ZAP, you can click **Help** on the Menu Bar or press F1 to access context-sensitive help from the ZAP User Guide.

For more information about the UI, see ZAP UI Overview in the ZAP online documentation.

ZAP also supports a powerful API and command line functionality, both of which are beyond the scope of this guide.

Launching Browsers

You can quickly and easily launch browsers that are pre-configured to proxy through ZAP via the Quick Start tab. Browsers launched in this way will also ignore any certificate validation warnings that would otherwise be reported.

This option will launch any of the most common browsers that you have installed with new profiles.

If you would like to use any of your browsers with an existing profile, for example with other browser add-ons installed, then you will need to manually configure your browser to proxy via ZAP and import and trust the ZAP Root CA Certificate. See the ZAP User Guide for more details.

Try to Connect Your Web Application

Once you have successfully set up your browser to use ZAP as its proxy, attempt to connect to the web application you are going to test.

If you are unable to reach your web application, check the following:

1. Verify the proxy settings the browser is using to connect to ZAP.
2. Verify the proxy settings in ZAP are those the browser is using to try to connect to ZAP.
3. Verify the web application you want to test is running.
4. Check to see whether your network requires a proxy to reach your web application. If so, you may need to configure ZAP to use a proxy.

To configure ZAP to use an outgoing proxy:

1. Start ZAP and on the Menu Bar, click **Tools** -> **Options**.
2. Select **Connection** in the left pane.
3. In the **use proxy chain** section of the **Connection** settings, check the **Use an outgoing proxy server checkbox**.
4. Enter the **Address/Domain Name** and **Port** for your network proxy.
5. Click **OK** to save the settings and verify that you can now connect to your web application.

Once your browser can successfully connect to your web application, you are ready to run a test.

Start Pentesting with ZAP

The easiest way to start using ZAP is to run a Quick Start test. Quick Start is a ZAP add-on that was installed automatically when you installed ZAP.

IMPORTANT: You should only use ZAP to attack an application you have permission to test with an active attack. Because this is a simulation that acts like a real attack, actual damage can be done to a site’s functionality, data, etc. If you are worried about using ZAP, you can prevent it from causing harm (though ZAP’s functionality will be significantly reduced) by switching to safe mode.

To switch ZAP to safe mode, click the arrow on the mode dropdown on the main toolbar to expand the dropdown list and select **Safe Mode**.

Run a Quick Start Test

To run a Quick Start test:

1. Start ZAP and click the **Quick Start** tab of the Workspace Window.
2. In the **URL to attack** text box, enter the full URL of the web application you want to attack.
3. Click the **Attack** button.

ZAP will proceed to crawl the web application with its spider, then passively scan each page it finds. Then ZAP will use the active scanner to attack all of the discovered pages, functionality, and parameters.

Interpret Your Test Results

As ZAP spiders your web application, it constructs a map of your web applications' pages and the resources used to render those pages. Then it records the requests and responses sent to each page and creates alerts if there is something potentially wrong with a request or response.

See Explored Pages

To examine a tree view of the explored pages, click the **Sites** tab in the Tree Window. You can expand the nodes to see the individual URLs accessed.

View Alerts and Alert Details

The left-hand side of the Footer contains a count of the Alerts found during your test, broken out into risk categories. These risk categories are:

To view the alerts created during your test:

1. Click the **Alerts** tab in the Information Window.
2. Click each alert displayed in that window to display the URL and the vulnerability detected in the right side of the Information Window.
3. In the Workspace Windows, click the **Response** tab to see the contents of the header and body of the response. The part of the response that generated the alert will be highlighted.

Expand Your Pentesting with ZAP

The passive scanning and automated attack functionality is a great way to begin a vulnerability assessment of your web application but it has some limitations. Among these are:

- Any pages protected by a login page are not discoverable during a passive scan because, unless you've configured ZAP's authentication functionality, ZAP will not handle the required authentication.
- Any pages that are not findable with ZAP's default spider are not testable during a passive scan. ZAP does provide additional options for discovery and coverage outside of passive scanning.
- You don't have a lot of control over the sequence of exploration in a passive scan or the types of attacks carried out in an automated attack. ZAP does provide many additional options for exploration and attacks outside of passive scanning.

Configure and Run a Spider with ZAP

One way to expand and improve your testing is to change the spider ZAP is using to explore your web application. Quick Scan uses the traditional ZAP spider, which discovers links by examining the HTML in responses from the web application. This spider is fast, but it is not always effective when exploring an AJAX web application that generates links using JavaScript.

For AJAX applications, ZAP's AJAX spider is likely to be more effective. This spider explores the web application by invoking browsers which then follow the links that have been generated. The AJAX spider is slower than the traditional spider and requires additional configuration for use in a "headless" environment.

A simple way to switch back and forth between spiders is to enable a tab for each spider in the Information Window and use that tab to launch scans.

1. In the Information Window, click the green plus sign (**+**).
2. Click **Spider** to create a Spider tab.
3. Repeat step 1, then click **AJAX Spider** to create an **AJAX Spider** tab.
4. Click the push-pin symbol on both the **Spider** and **AJAX Spider** tabs to pin them to the Information Window.

Note that both of these tabs include a **New Scan** button.

Explore Your Site

Spiders are a great way to explore your basic site, but they should be combined with manual exploration to be more effective. Spiders, for example, will only enter basic default data into forms in your web application but a user can enter more relevant information which can, in turn, expose more of the web application to ZAP. This is especially true with things like registration forms where a valid email address is required. The spider may enter a random string, which will cause an error. A user will be able to react to that error and supply a correctly formatted string, which may cause more of the application to be exposed when the form is submitted and accepted.

Since you have configured your browser to use ZAP as its proxy, you should explore all of your web application with that browser. As you do this, ZAP passively scans all the requests and responses made during your exploration for vulnerabilities, continues to build the site tree, and records alerts for potential vulnerabilities found during the exploration.

It is important to have ZAP explore each page of your web application, whether linked to another page or not, for vulnerabilities. Obscurity is not security, and hidden pages sometimes go live without warning or notice. So be as thorough as you can when exploring your site.

Run an Active Scan with ZAP

So far ZAP has only carried out passive scans of your web application. Passive scanning does not change responses in any way and is considered safe. Scanning is also performed in a background thread to not slow down exploration. Passive scanning is good at finding some vulnerabilities and as a way to get a feel for the basic security state of a web application and locate where more investigation may be warranted.

Active scanning, however, attempts to find other vulnerabilities by using known attacks against the selected targets. Active scanning is a real attack on those targets and can put the targets at risk, so do not use active scanning against targets you do not have permission to test.

To start an active scan:

1. In the Tree View, in the **Sites** tab, select the sites you want to perform an active scan on.
2. Right-click the selected sites and select **Active Scan**.

or

1. In the Information Window, select the **Active Scan** tab.
2. Click **New Scan**.

To review and modify your settings, then begin an active scan:

1. In the Menu Bar, click **Tools** -> **Active Scan**.
2. Review the settings and make any changes you wish to.
3. Click **Start Scan** to start the Active Scan with these settings.

You can review the results of your active scan the same way you reviewed the results of your passive scan, as shown in Interpret Your Test Results.

Learn More About ZAP

Now that you are familiar with a few basic capabilities of ZAP, you can learn more about ZAP's capabilities and how to use them from ZAP's User Guide. The User Guide provides step-by-step instructions, references for the API and command-line programming, instructional videos, and tips and tricks for using ZAP.

Useful Links

OWASP Zed Attack Proxy Project – ZAP's main project page

OWASP ZAP Wiki – The ZAP Wiki

OWASP ZAP User Guide - The ZAP User Guide

OWASP ZAP Hot Keys - The list of ZAP hotkeys

ZAP Users Group – Google group for ZAP users

ZAP Developers Group – Google group for developers and contributors to ZAP