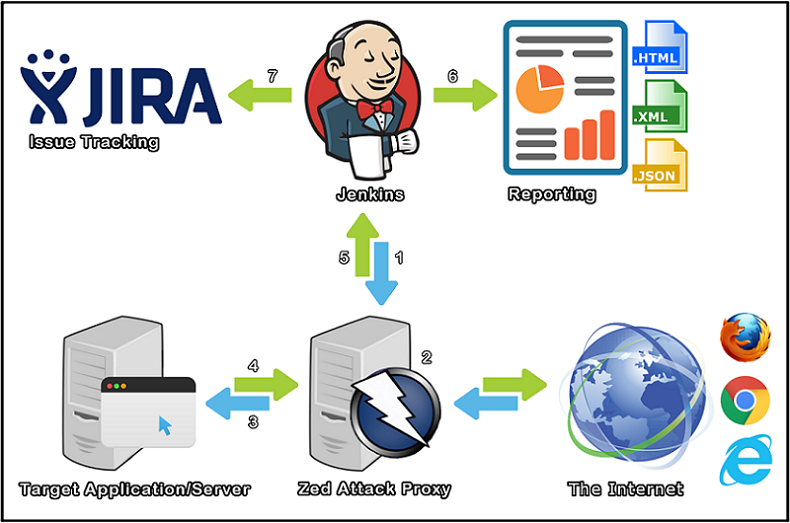
Integration of ZAP Plugin with Jenkins

# OWASP ZAP (Open Source)

**Overview of Jenkins – ZAP integration**

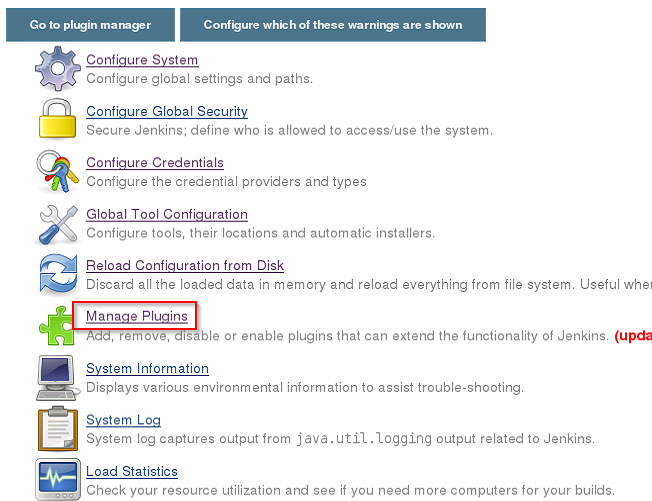


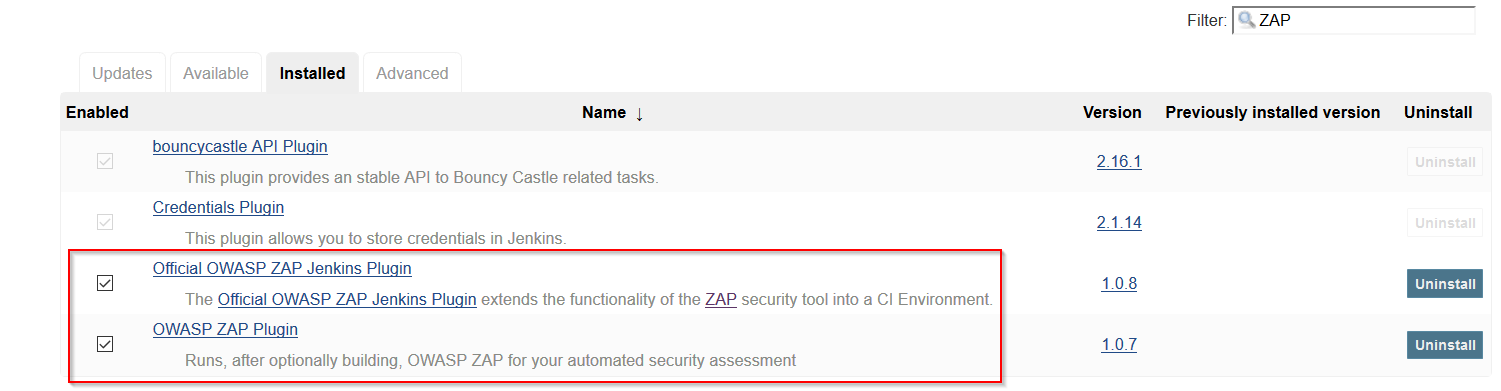
1. A Jenkins CI Build step initializes ZAP
2. Traffic flows (Regression Pack) through ZAP (Web Proxy)
3. ZAP modifies requests to include Vulnerability Tests
4. Target Application/Server sends Response back through ZAP
5. ZAP sends reporting data back to Jenkins
6. Jenkins publishes and archives the report(s)
7. Jenkins creates JIRA tickets for the alerts

**Deployment & Integration Steps:**

1. **Installation of OWASP ZAP and custom tool setup**
2. Click on “**Manage Jenkins**” option and go to “**Manage Plugins**” and click on the available tab and check for “**Official OWASP ZAP Jenkins Plugin**” in the search bar and Click the option “**Install without restart**” as shown in the following screenshots.





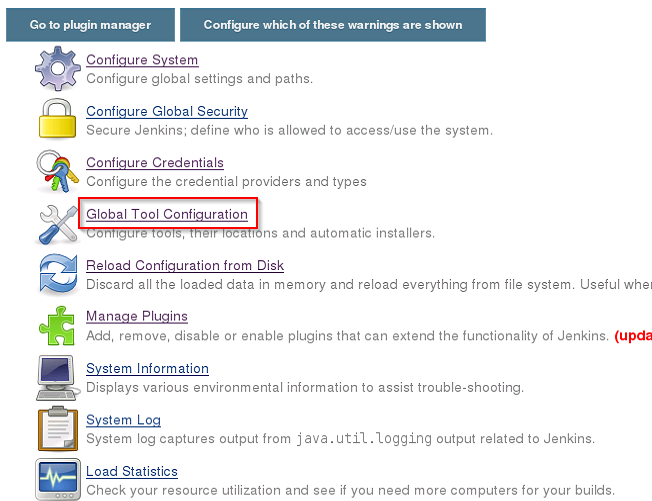


1. **Configure ZAP as custom tool**

This section defines the installation of the ZAP tool within Jenkins as a custom tool.

1. Go to **Manage Jenkins- > Global Tool**



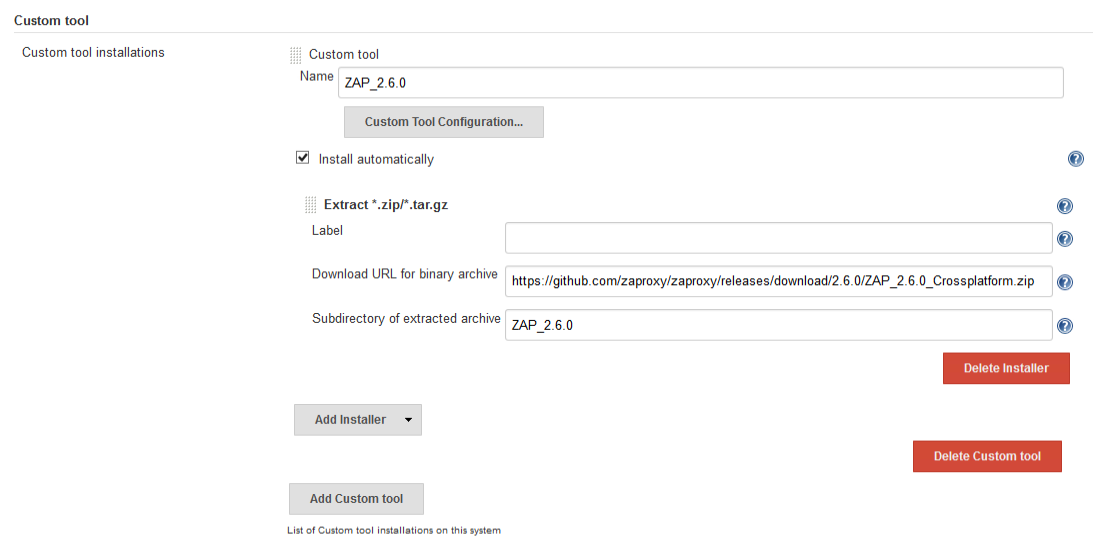


1. Scroll down to “**Custom tool”** section and click on “**Custom tool installations**”.
2. In the custom tool section, following fields should be populated. Select the checkbox “**Install automatically**”
3. “**Name**” = ZAP\_2.6.0 (this can be a custom name given to the ZAP tool)
4. “**Download URL for binary usage”** = Depending on the operating system. <https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Linux.tar.gz>

<https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2.6.0_Crossplatform.zip>

<https://github.com/zaproxy/zaproxy/releases/download/2.6.0/ZAP_2_6_0_windows.exe>

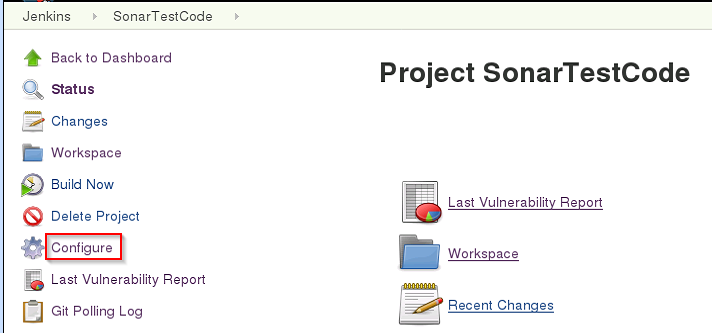
1. “**Subdirectory of extracted archive**” = Mention a custom folder name

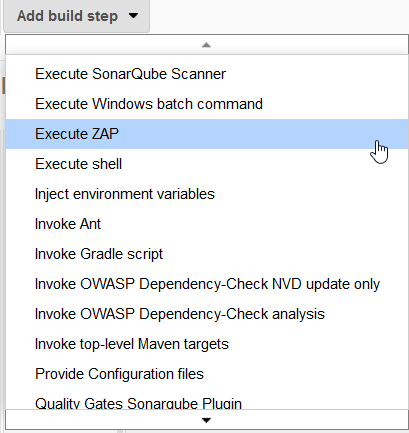


1. **Adding the OWASP ZAP Plugin:**

This plugin uses the ZAP tool installed under the custom tool **(in Section B above)** and executes based on the configuration made in the plugin.

1. Click on “**Configure**” in the left navigation panel and navigate to the “**Build**” tab and add the “**Execute ZAP**” plugin as shown below.

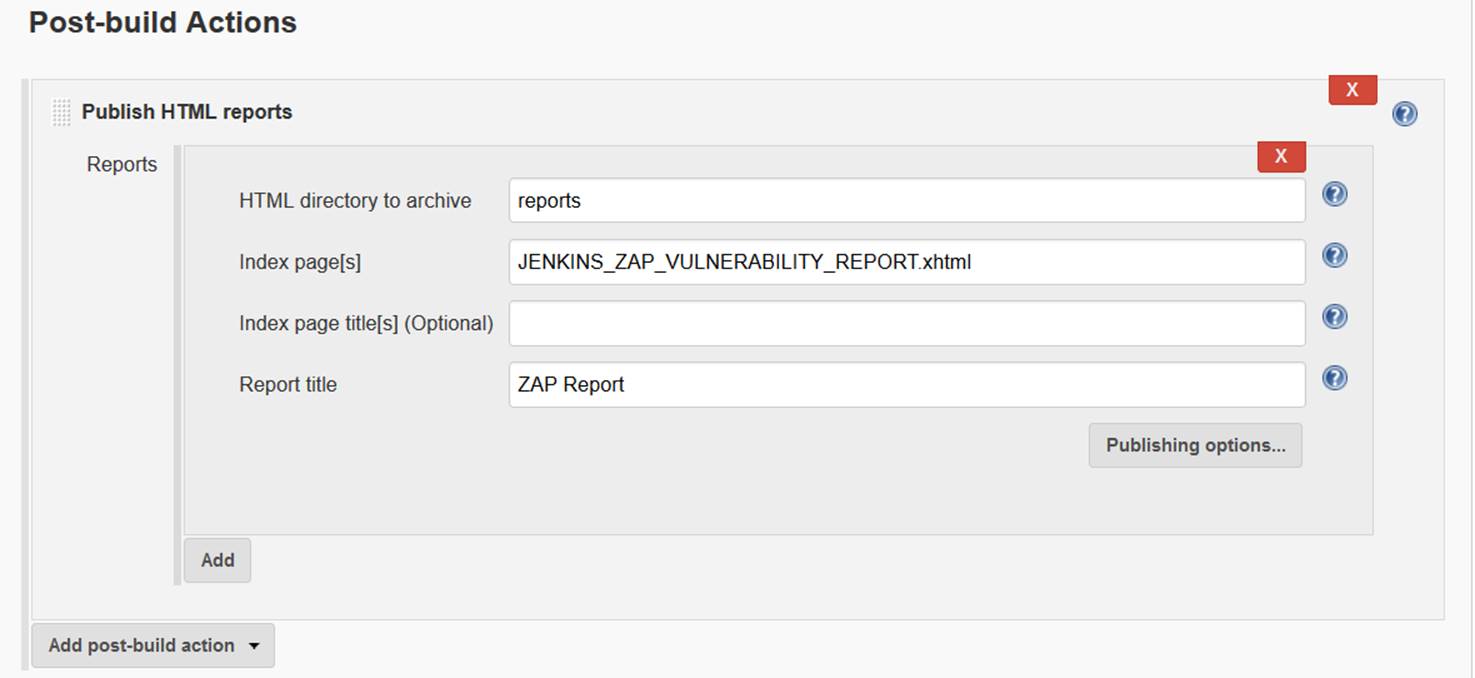




1. The table below provides the configuration sections under the “**Build**” section, that should be populated to achieve ZAP integration with Jenkins.

|  |  |
| --- | --- |
| **Sub Section Name** | **Configurations to be performed** |
| Admin Configurations | Override Host – **127.0.0.1** Override Port – **8090** |
| Installation Method | Select “**Custom Tools Installation**” and select the ZAP tool name from the drop down in the “**Name**” field. |
| ZAP Settings | Set the default folder path of the ZAP tool to **C:\users\%username%\OWASP ZAP** |
| Session Management | Select “**Persist Session**” and enter the filename as “**sessions**” |
| Session Properties | Enter the “**Context Name**” as the application name (“**testfire**”). Enter the URL’s to be included in “**Include in Context**” field (“**http://testfire.net/\***”)    Enter the credentials of the application as shown below: |
| Attack Mode | Update the URL of the application under the “**Attack Mode”** and select the mode of scan as shown in the below screenshot. |
| Final Run | Select the type of output report that is required by selecting the option “**Generate Report**” and fill all the necessary details as shown in the below screenshot. |

1. Add the “**Publish HTML reports**” under the Post Build Actions as an additional functionality to export the report separately.

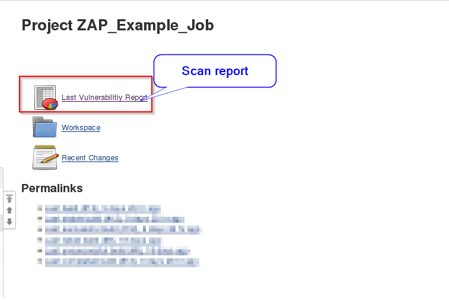


1. Click **Apply** & **Save**.

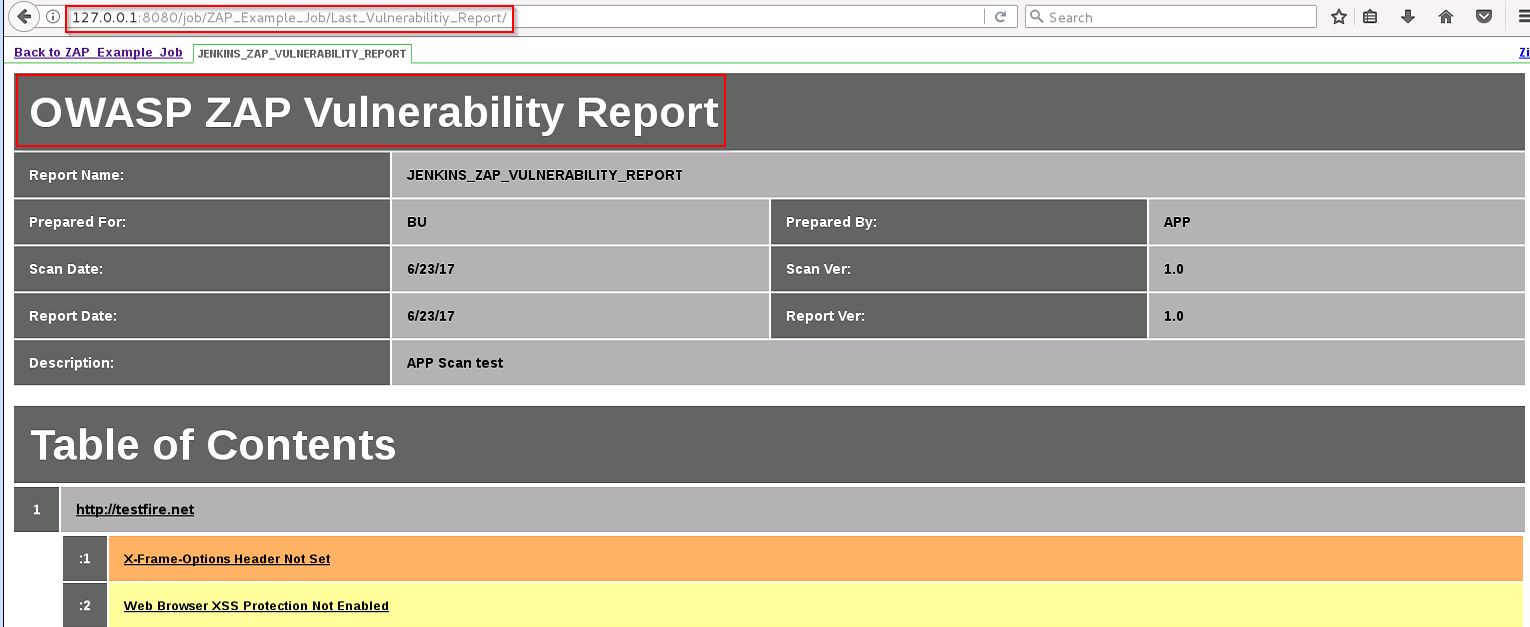


**Scanning & Reporting Steps:**

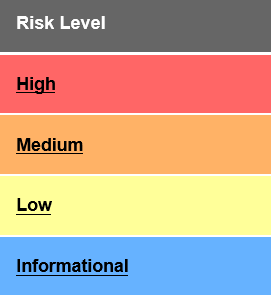
1. **Vulnerability Scanning**
   1. Based on the value set in the “**Section D Step 3”** for **Poll SCM Schedule**, the build will be triggered. This inturn invokes the OWASP ZAP plugin and executes.
2. **Generate Scan Report**
   1. Once the build is successful, the scan report is available in the project Jenkins dashboard.



1. Click on the report to see the reported vulnerabilities in detail.



1. **Build Pass/Fail & Remediation Guidelines**



|  |  |
| --- | --- |
| **Build Pass Criteria** | Zero High Risk Level Vulnerabilities\* |
| **Build Fail Criteria** | High Risk Level > 1 |

\* Any false positives identified by the development team in the report, should be discussed with the Security team for suppression of false positive vulnerabilities