ATTACK DEFENSE

by PentesterAcademy

Name	DevSecOps Pipeline: Java WebApp
URL	https://www.attackdefense.com/challengedetails?cid=2066
Type	Pipeline Basics: Web Applications

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

DevOps practices are to combine software development (Dev) and IT operations (Ops) in order to improve the delivery process. DevOps pipelines are chained tasks and components that run in a sequence to cover different phases of software compilation, packaging, automated testing, and test deployment.

In this lab, we have a simple DevOps pipeline for a sample Java-based web application. The pipeline consists of the following components (and tasks):

- Kali machine (For pulling, modifying, and pushing the code)
- GitLab server (For hosting code)
- Jenkins server (For integrating). Different phases and components used:
    - Build: Maven
    - Code testing: Maven
    - Test Deployment: Ansible
    - Dynamic Testing: Selenium
- Test server (For test deployment)

It is suggested to play the DevOps focused lab before playing this lab.

DevSecOps refer to introducing security in different stages of the DevOps process. This is done to catch the vulnerabilities/insecurities as soon as possible in the pipeline. In this lab, the pipeline consists of the following components (and tasks):

- Sensitive Information Scan phase: Talisman
- Software Component Analysis: OWASP dependency check
- Dynamic Application Security Testing: OWASP ZAP
- Vulnerability Assessment: OpenVAS

Objective: Run the pipeline and observe/understand the DevSecOps process!

Instructions:

- The GitLab server is reachable with the name 'gitlab'
- Gitlab credentials:

Username	**Password**
root	welcome123

- The Jenkins server is reachable with the name 'jenkins'
- Jenkins credentials:

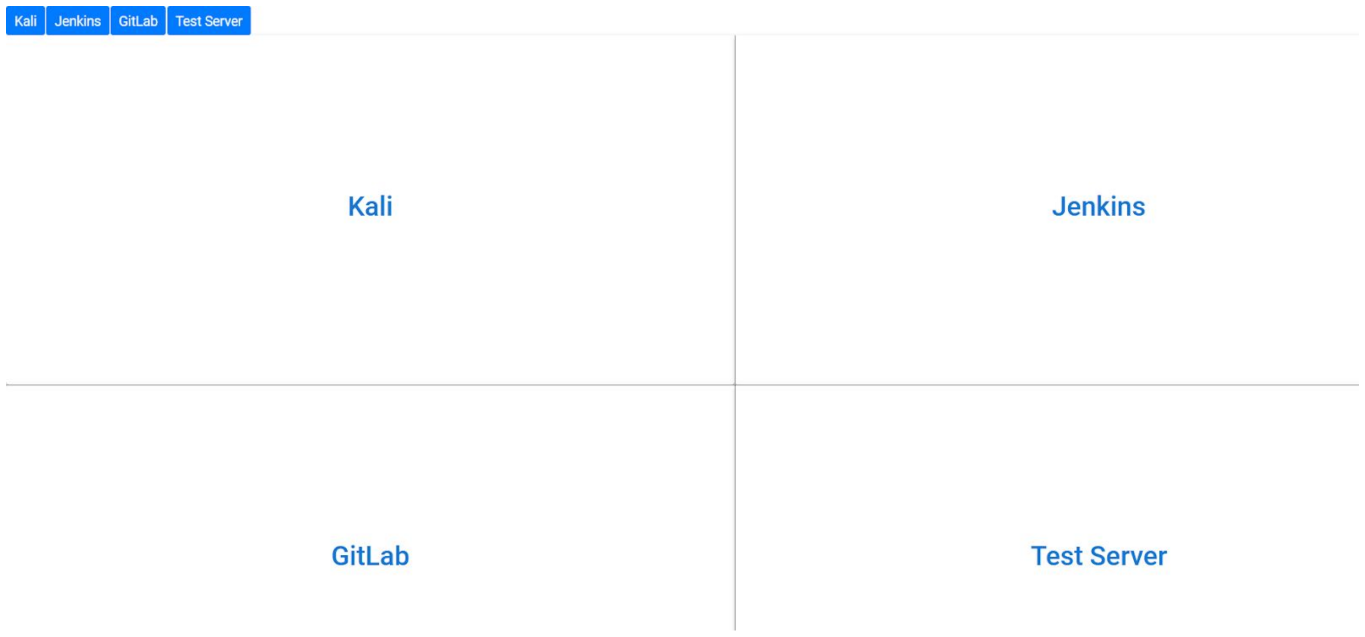
Username	**Password**
admin	welcome123

- The test deployment server is reachable by the name "test-server"
- Test server SSH credentials:

Username	**Password**
tomcat	password1

Lab Setup

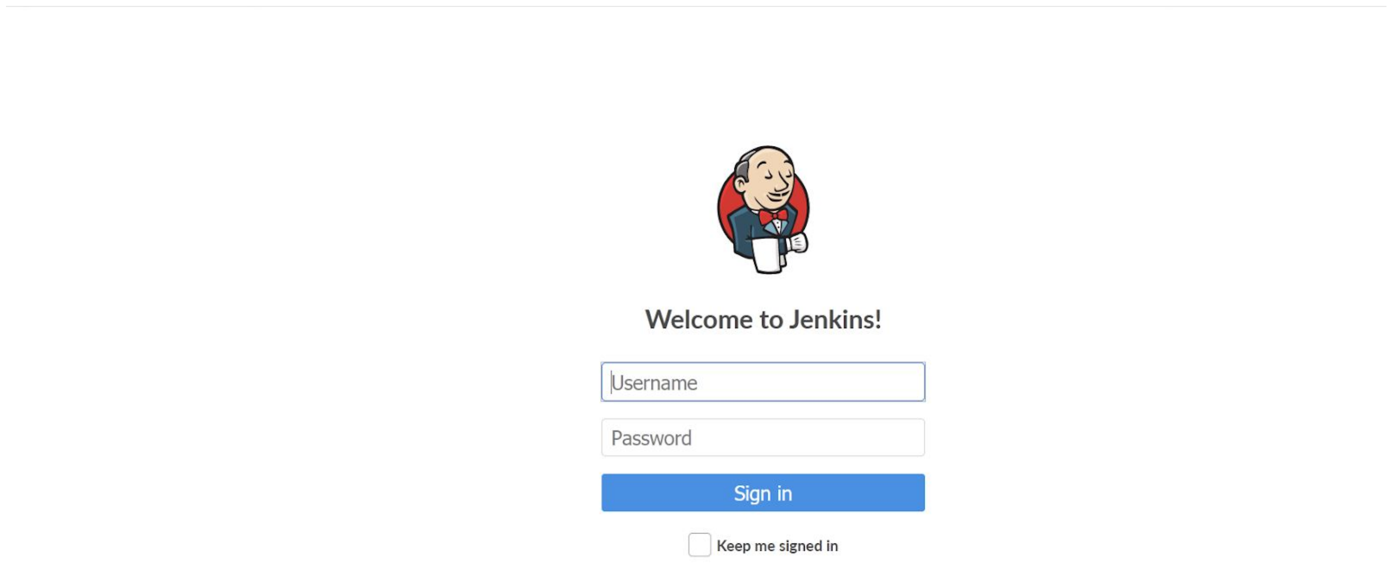
On starting the lab, the following interface will be accessible to the user.



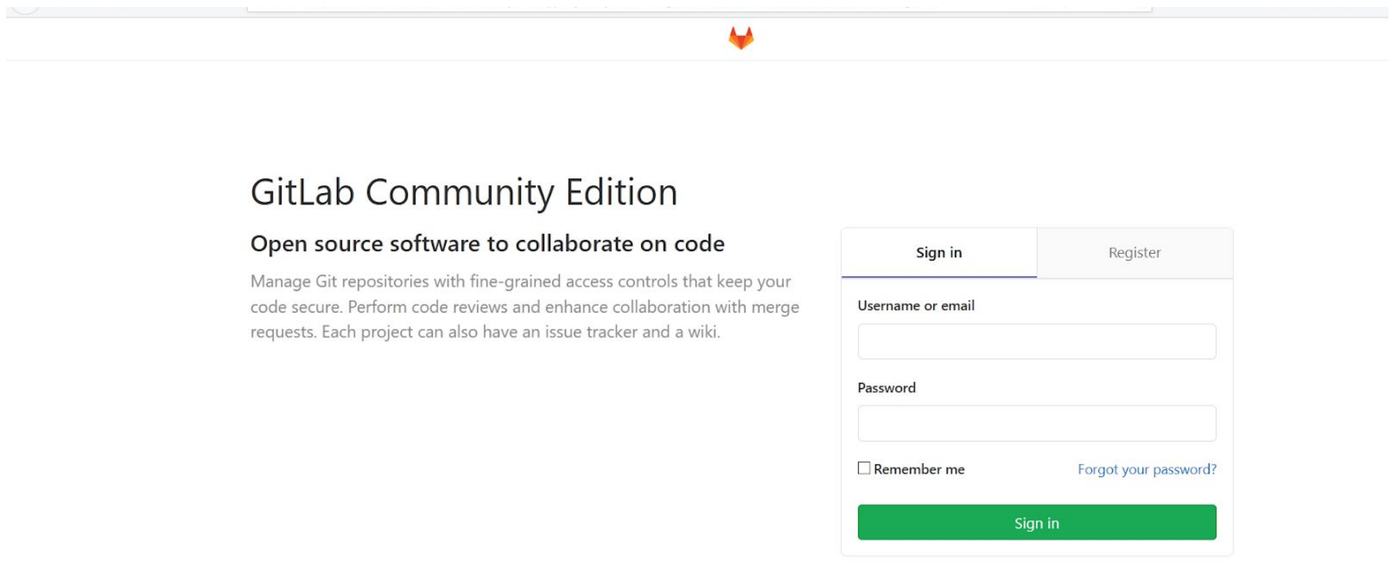
On choosing (clicking the text in the center) top left panel, **KALI CLI** will open in a new tab

```
root@kali-cli:~#
```

Similarly on selecting the top right panel, a web UI of **Jenkins** will open in a new tab.



On selecting the bottom left panel, a web UI of **Gitlab** will open in a new tab.


GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

And on selecting the bottom right panel, a web UI of **Test Server** will open in a new tab.

PENTESTER
ACADEMY
WebApp will appear once deployed!
This page refreshes automcatically.

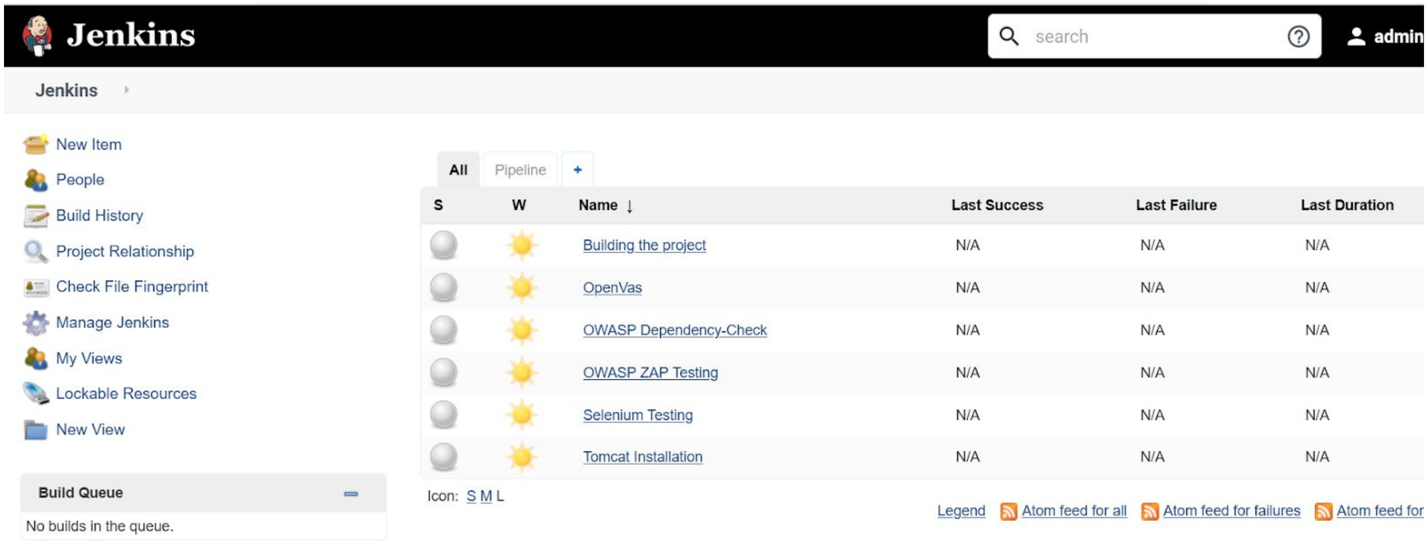
The page will reload until the test-server has started running the web service at port 8080

Solution

Step 1: Login into the Jenkins, The credentials are provided in the challenge description.

Credentials:

- **Username:** admin
- **Password:** welcome123


Jenkins

search

admin

Jenkins

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Lockable Resources
- New View

All | Pipeline | +

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Building the project	N/A	N/A	N/A
		OpenVas	N/A	N/A	N/A
		OWASP Dependency-Check	N/A	N/A	N/A
		OWASP ZAP Testing	N/A	N/A	N/A
		Selenium Testing	N/A	N/A	N/A
		Tomcat Installation	N/A	N/A	N/A

Build Queue

No builds in the queue.

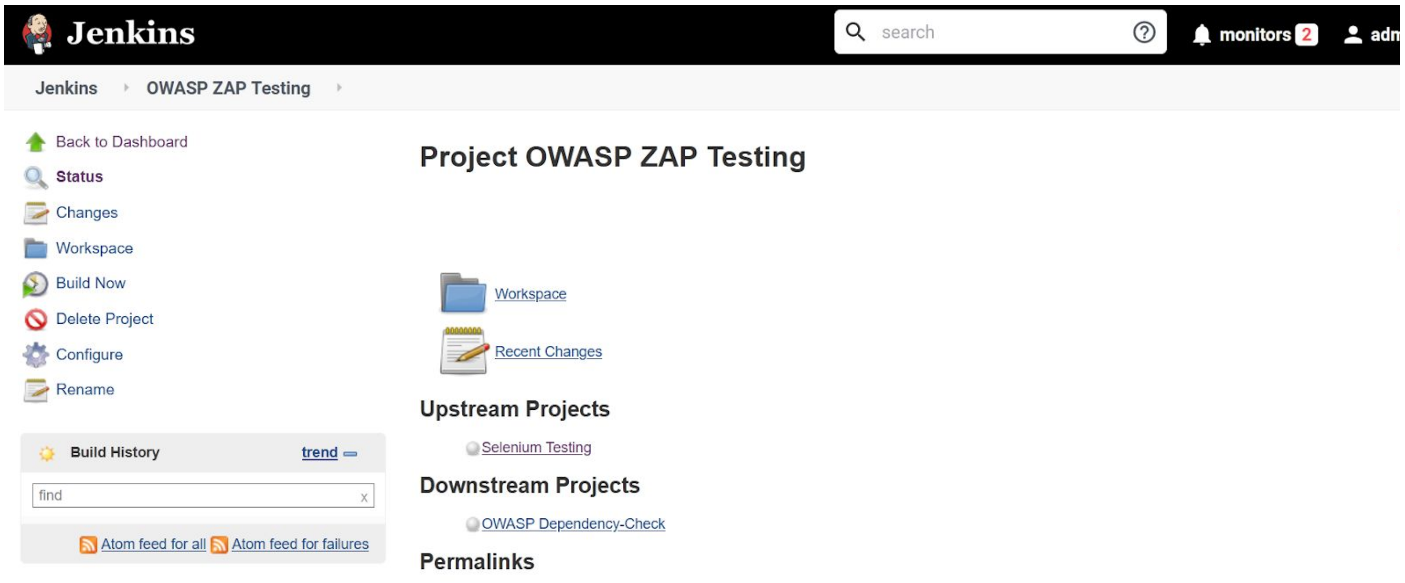
Icon: S M L

Legend | Atom feed for all | Atom feed for failures | Atom feed for

There are 6 jobs present in the Jenkins Interface, We will take one job (DevSecOps only) at a time to study as DevOps jobs are already covered in the DevOps pipeline lab. Please check that first if you haven't already.

Job 1: OWASP ZAP Testing

Step 1: Click on the "OWASP ZAP Testing" job.


Jenkins

search

monitors 2

Jenkins > OWASP ZAP Testing

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- Rename

Build History trend

find

Atom feed for all | Atom feed for failures

Project OWASP ZAP Testing

Workspace

Recent Changes

Upstream Projects

Selenium Testing

Downstream Projects

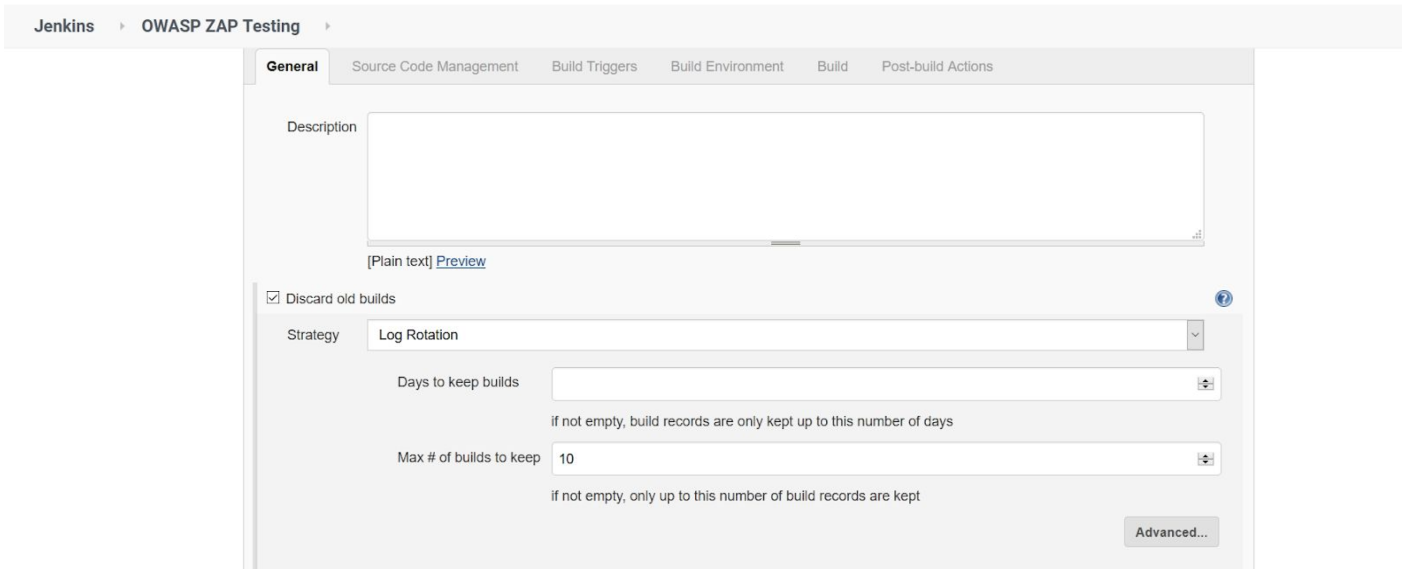
OWASP Dependency-Check

Permalinks

This is the console for "OWASP ZAP Testing" job, from here the user can configure, Build, Delete the Job.

The workspace is a personal directory of this job where all the files which get downloaded during the build phase will reside.

Step 2: Click on the "Configure" option to check the configuration of the Job.



Description

[Plain text] Preview

☑ Discard old builds

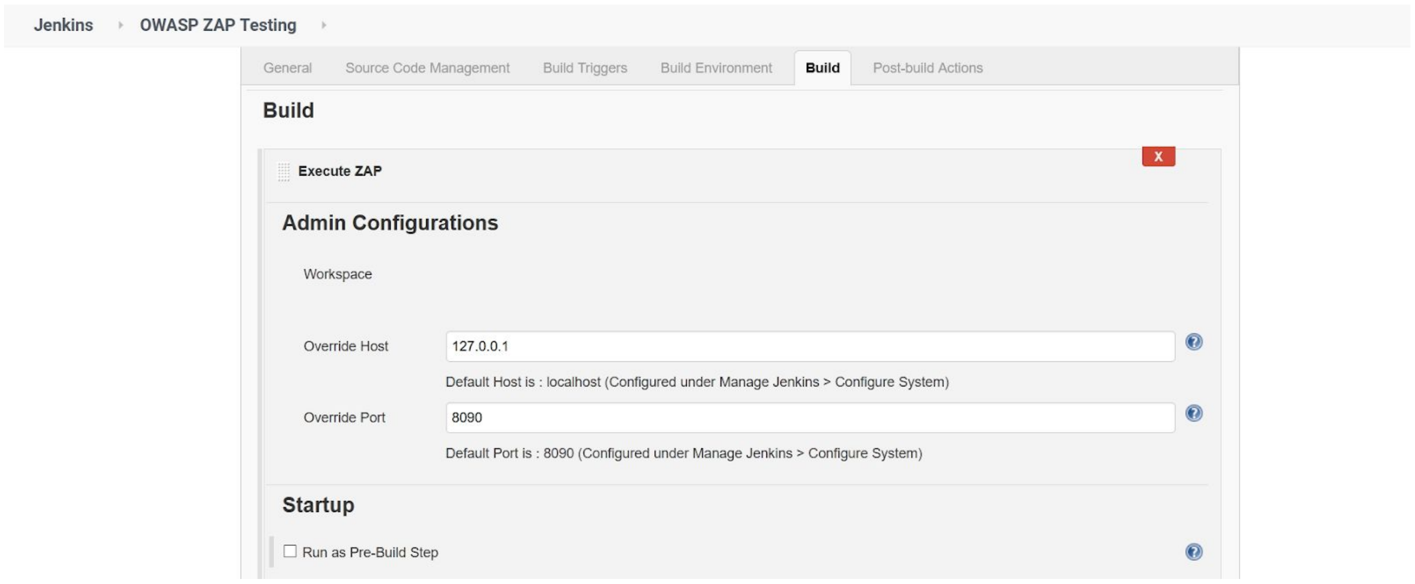
Strategy: Log Rotation

Days to keep builds:

if not empty, build records are only kept up to this number of days

Max # of builds to keep: 10

if not empty, only up to this number of build records are kept



Build

Execute ZAP

Admin Configurations

Workspace

Override Host: 127.0.0.1

Default Host is : localhost (Configured under Manage Jenkins > Configure System)

Override Port: 8090

Default Port is : 8090 (Configured under Manage Jenkins > Configure System)

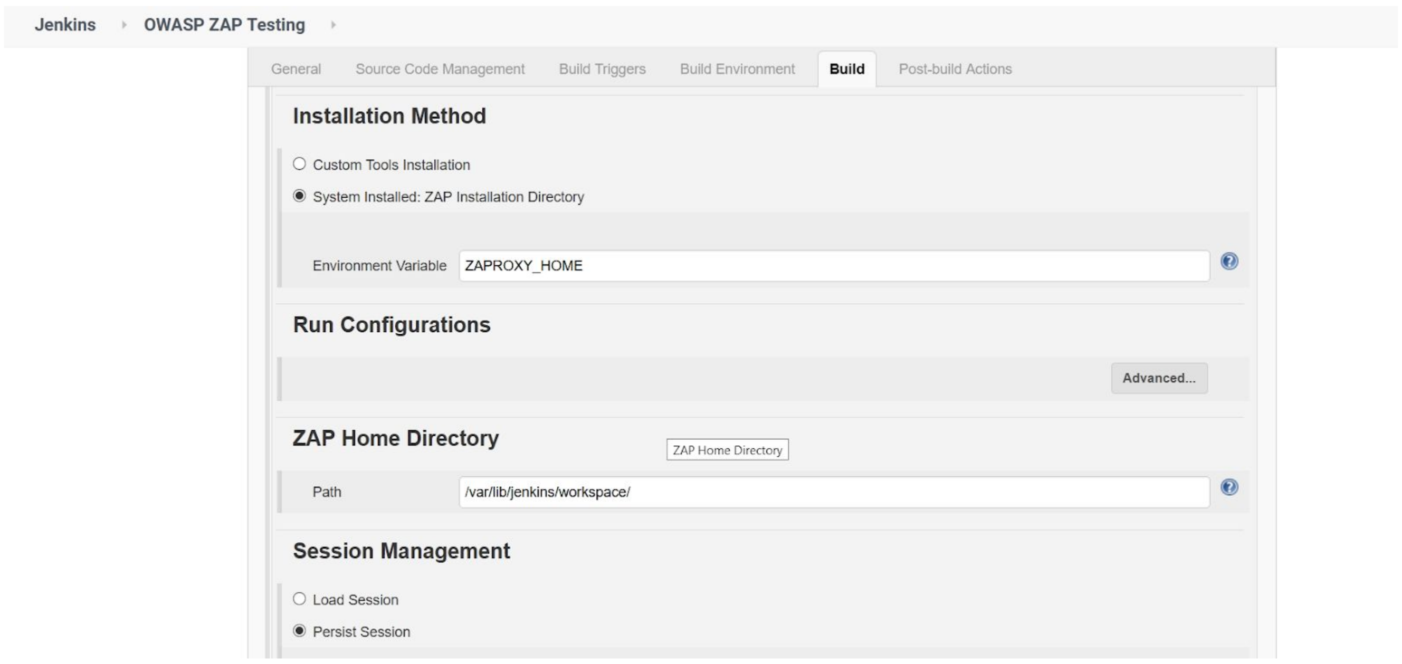
Startup

☐ Run as Pre-Build Step

The Log Rotation is enabled on this job which means after every 10 rotations of the build phase, the logs will be deleted.

This phase is the "Build" phase where the application gets compiled or packages, The OWASP ZAP plugin is accepting 2 parameters (Host and port). The host and port should be of the OWASP ZAP server. Which in our case is running at port 8090 of the localhost.

Step 3: Scroll down to check more options for the Job configuration



Installation Method

- ○ Custom Tools Installation
- ● System Installed: ZAP Installation Directory

Environment Variable: ZAPROXY_HOME

Run Configurations

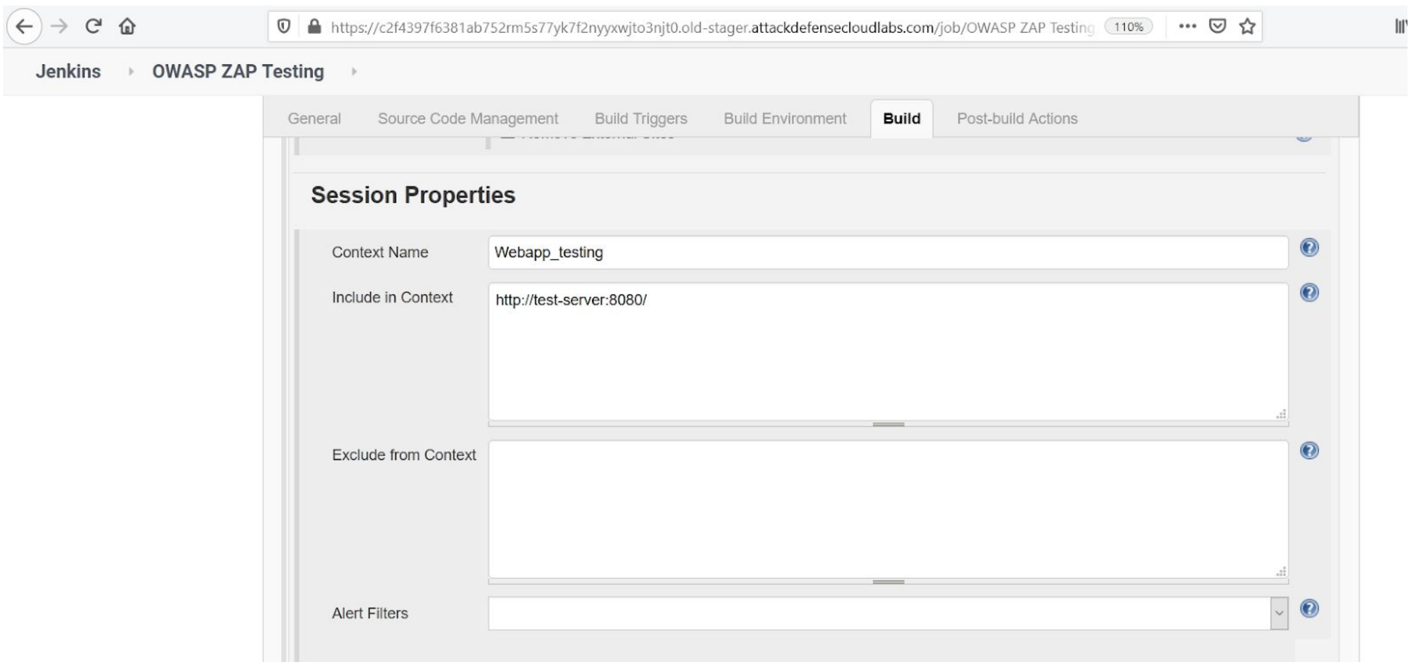
ZAP Home Directory

Path: /var/lib/jenkins/workspace/

Session Management

- ○ Load Session
- ● Persist Session

Under the Installation method, There are 2 options to choose for OWASP ZAP's installation directory. The "ZAPROXY_HOME" is the environment variable which points to the installation directory of the OWASP ZAP tool.



https://c2f4397f6381ab752rm5s77yk7f2nyyxwjto3njt0.old-stager.attackdefensecloudlabs.com/job/OWASP ZAP Testing

Jenkins › OWASP ZAP Testing ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Session Properties

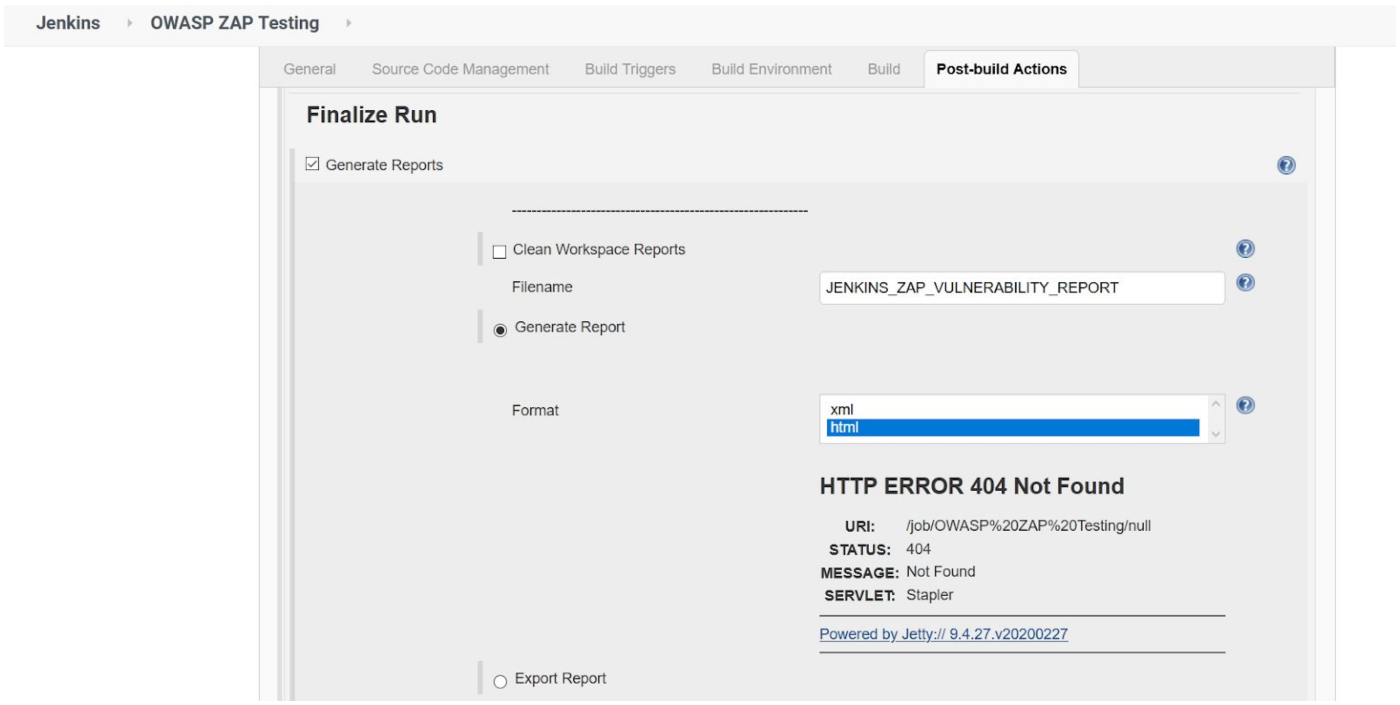
Context Name: Webapp_testing

Include in Context: http://test-server:8080/

Exclude from Context:

Alert Filters:

The plugin requires the name and the context URL to perform the scanning on the target application.


Jenkins › OWASP ZAP Testing ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Finalize Run

☑ Generate Reports

☐ Clean Workspace Reports

Filename: JENKINS_ZAP_VULNERABILITY_REPORT

◉ Generate Report

Format: xml, html (html selected)

HTTP ERROR 404 Not Found

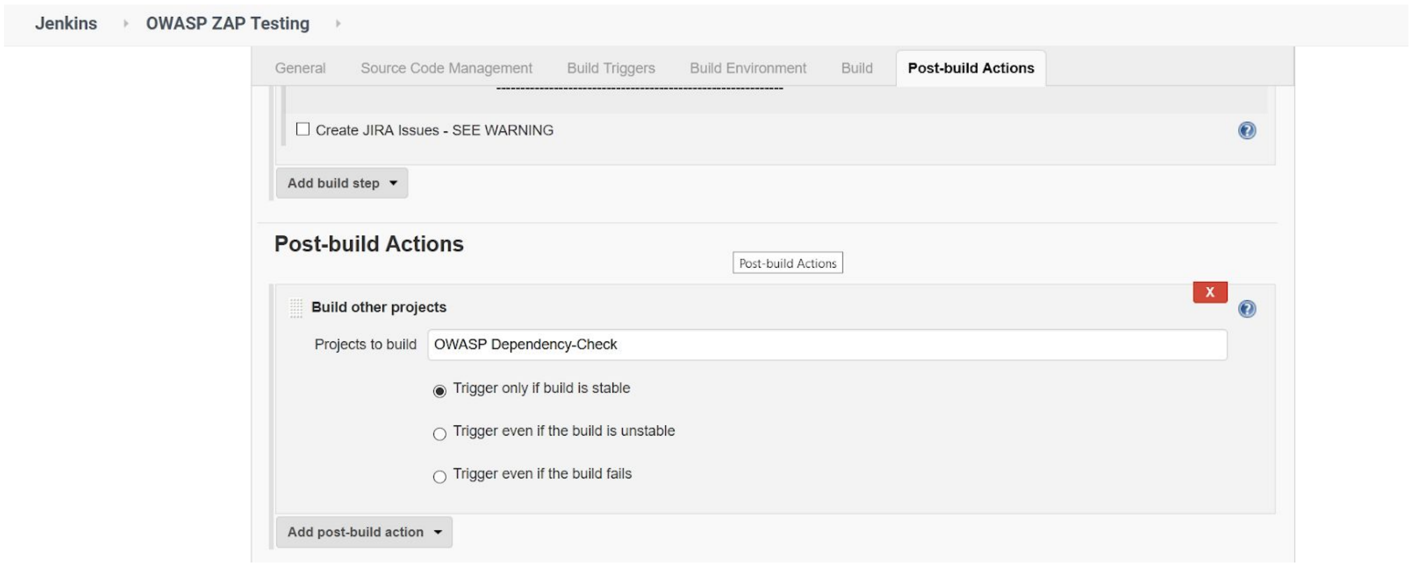
```
URI:     /job/OWASP%20ZAP%20Testing/null
STATUS:  404
MESSAGE: Not Found
SERVLET: Stapler
```

Powered by Jetty:// 9.4.27.v20200227

○ Export Report

After the scan is completed, the Plugin will generate a report based on the findings.

Post Build Actions: These are the steps which get executed after performing the Build phase. Here, the action is to execute another Job (OWASP Dependency Check) after finishing this one.


Jenkins › OWASP ZAP Testing ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

☐ Create JIRA Issues - SEE WARNING

Add build step

Post-build Actions

Build other projects

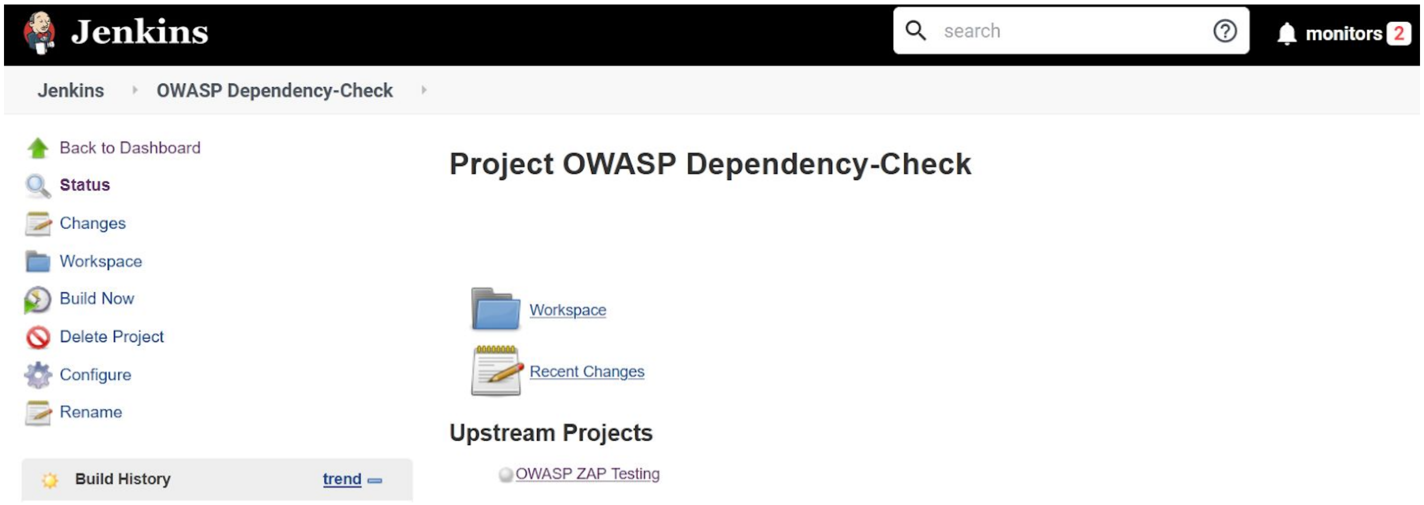
Projects to build: OWASP Dependency-Check

[✓ Trigger only if build is stable]
○ Trigger even if the build is unstable
○ Trigger even if the build fails

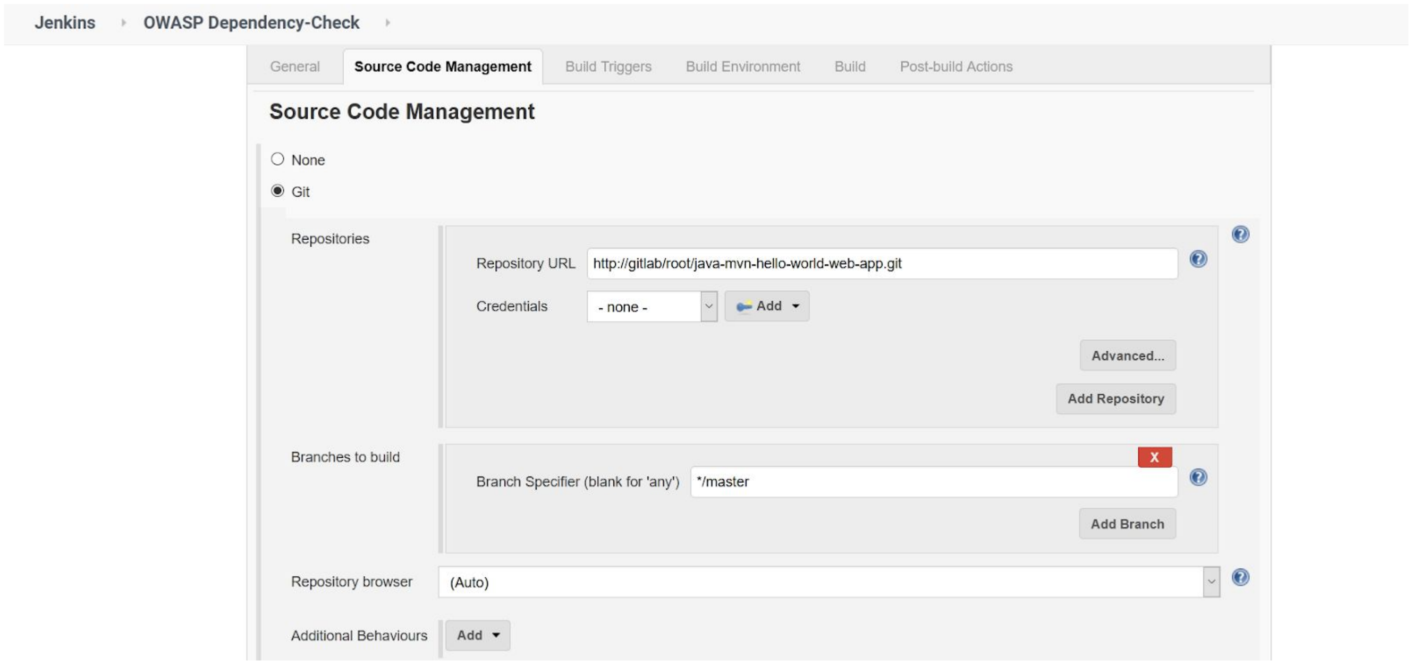
Add post-build action

Job 2: OWASP Dependency-Check Scan

Step 1: Click on the “OWASP Dependency check” job.



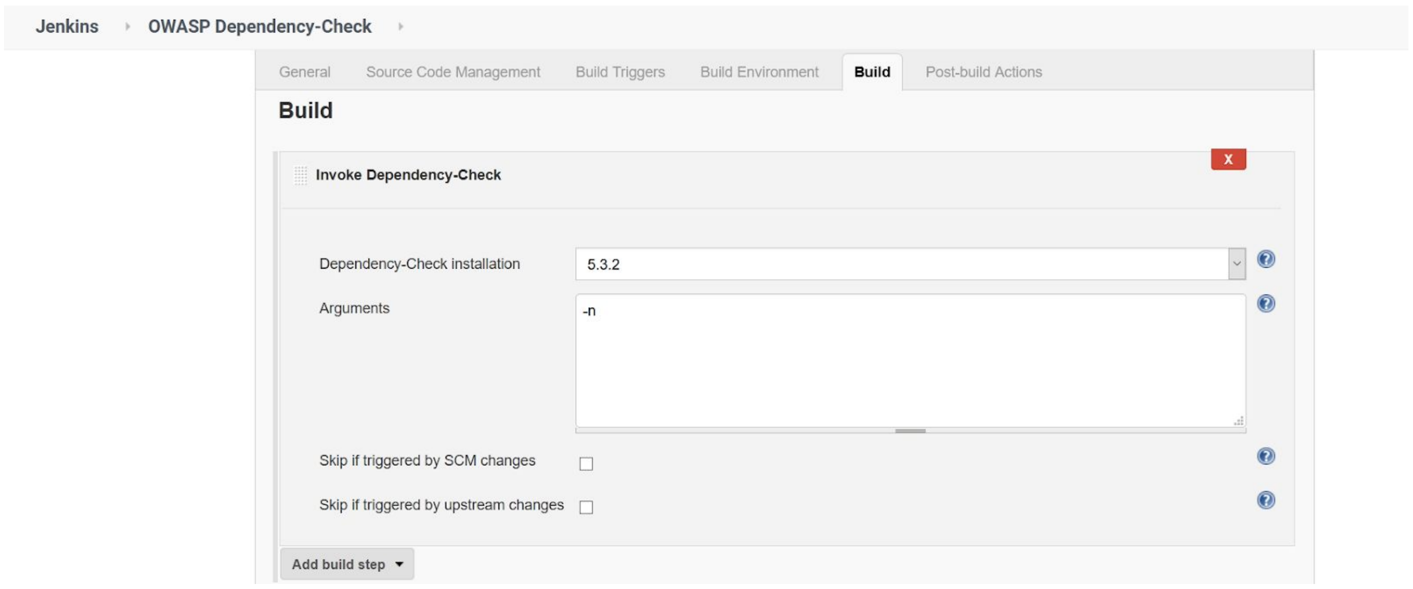
Step 2: Click on the “Configure” option to check the configuration of the Job.



Source Code Management: Git

Repository URL	http://gitlab/root/java-mvn-hello-world-web-app.git
Credentials	- none -
Branch Specifier (blank for 'any')	*/master
Repository browser	(Auto)
Additional Behaviours	Add

The Source Code Management (SCM) is the repository where the source code of the application is stored. This option will tell Jenkins to pull the code from the provided repository



Build

Invoke Dependency-Check

Dependency-Check installation: 5.3.2

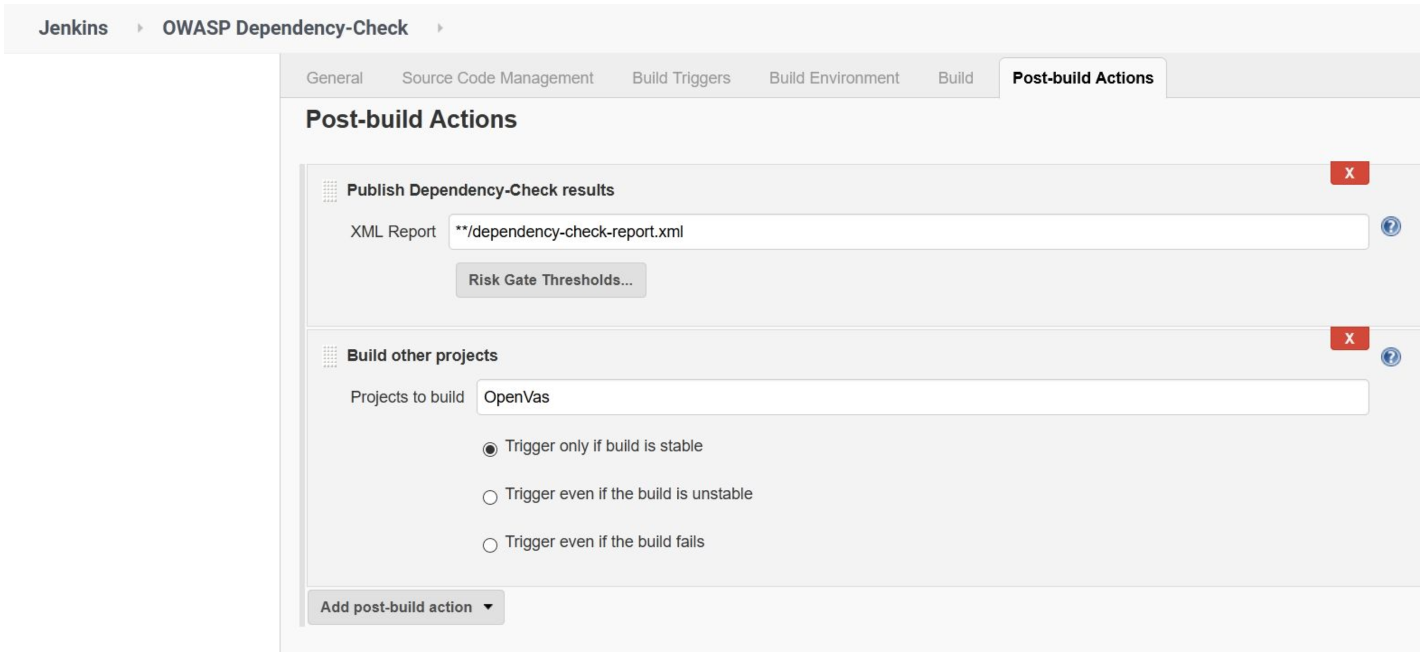
Arguments: -n

Skip if triggered by SCM changes ☐

Skip if triggered by upstream changes ☐

The OWASP Dependency-Check has a plugin which will be used to scan the application. The '-n' parameter passed as argument ensures that the tool will not check for any updates while scanning. Since the lab is running with no internet so checking for updates could cause issues.

Step 3: Check the Post-build Section from the Job configuration


Jenkins › OWASP Dependency-Check ›

Post-build Actions

Publish Dependency-Check results

XML Report: **/dependency-check-report.xml

Build other projects

Projects to build: OpenVas

[✓ Trigger only if build is stable]

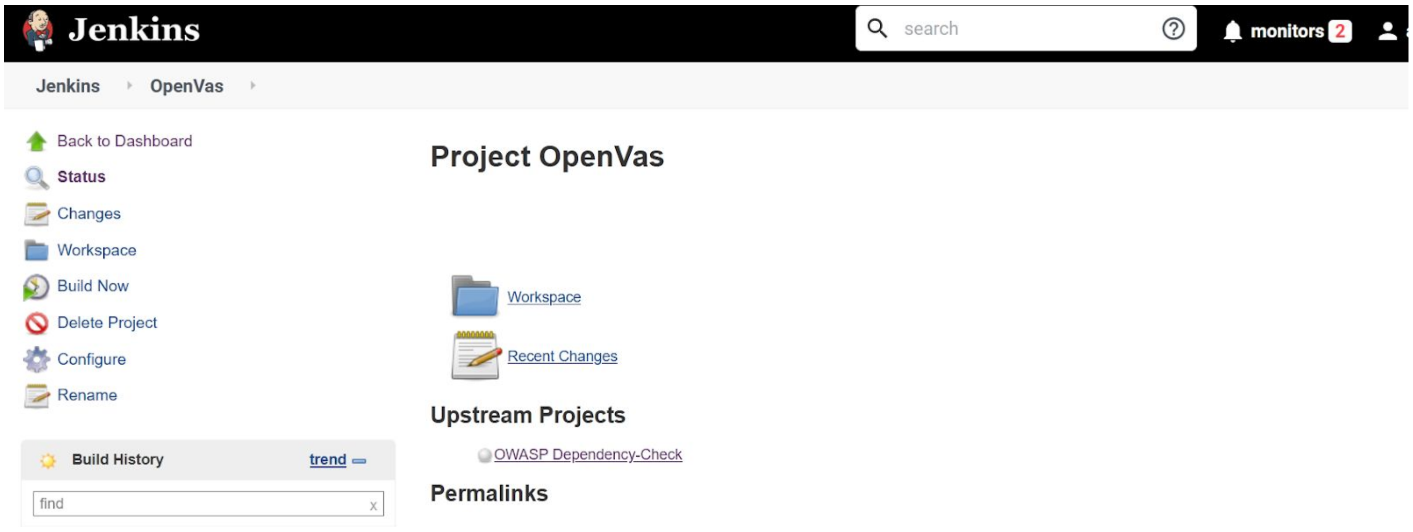
Trigger even if the build is unstable

Trigger even if the build fails

After the scan completes, an XML report will be generated and placed in the workspace of the Job. The next Job (OpenVas) will be initiated after completing this job.

Job 3: OpenVas

Step 1: Click on the "OpenVas" job.


Jenkins

search

monitors 2

Jenkins › OpenVas ›

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- Rename

Build History trend

find

Project OpenVas

Workspace

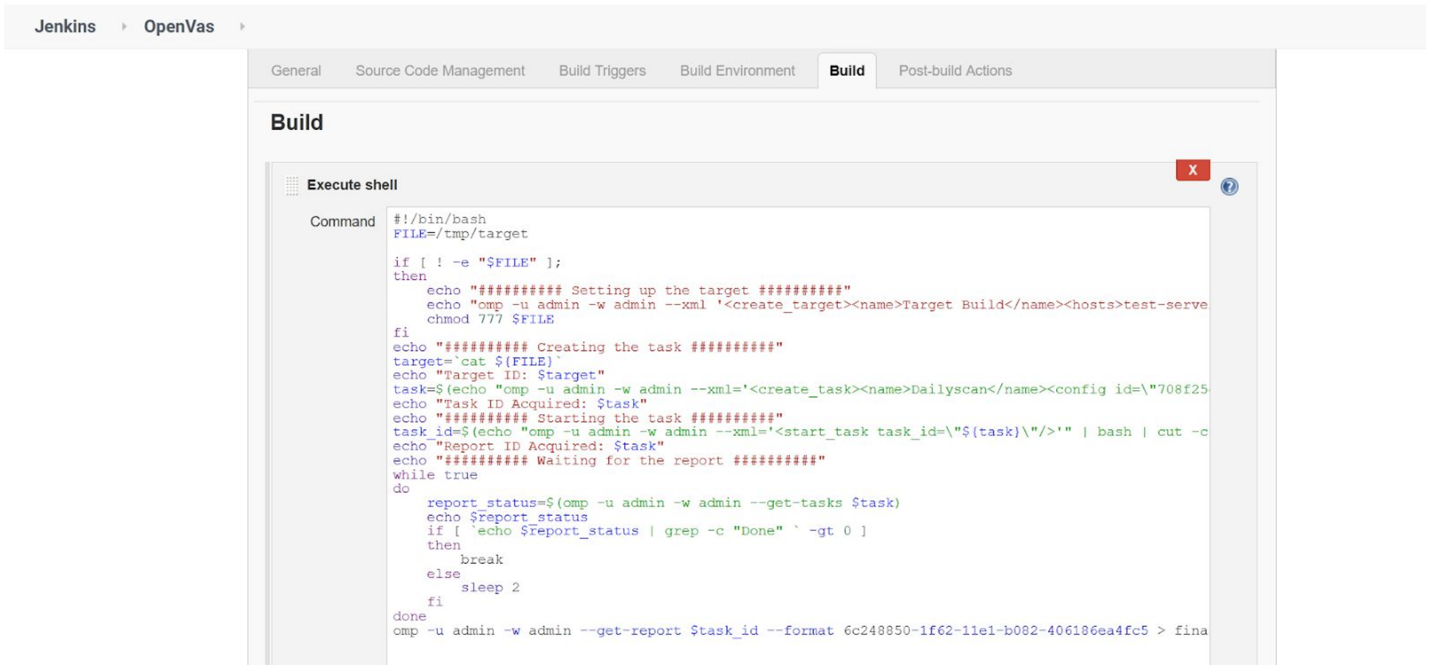
Recent Changes

Upstream Projects

OWASP Dependency-Check

Permalinks

Step 2: Click on the “Configure” option to check the configuration of the Job.


Jenkins › OpenVas ›

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Build

Execute shell

Command

```
#!/bin/bash
FILE=/tmp/target

if [ ! -e "$FILE" ];
then
    echo "########## Setting up the target ##########"
    echo "omp -u admin -w admin --xml '<create_target><name>Target Build</name><hosts>test-serve
    chmod 777 $FILE
fi
echo "########## Creating the task ##########"
target=`cat ${FILE}`
echo "Target ID: $target"
task=$(echo "omp -u admin -w admin --xml='<create_task><name>Dailyscan</name><config id=\"708f25
echo "Task ID Acquired: $task"
echo "########## Starting the task ##########"
task_id=$(echo "omp -u admin -w admin --xml='<start_task task_id=\"${task}\"/>'" | bash | cut -c
echo "Report ID Acquired: $task"
echo "########## Waiting for the report ##########"
while true
do
    report_status=$(omp -u admin -w admin --get-tasks $task)
    echo $report_status
    if [ `echo $report_status | grep -c "Done" ` -gt 0 ]
    then
        break
    else
        sleep 2
    fi
done
omp -u admin -w admin --get-report $task_id --format 6c248850-1f62-11e1-b082-406186ea4fc5 > fina
```

A custom Wrapper is implemented under the Build section to utilise the OpenVas using the omp binary.

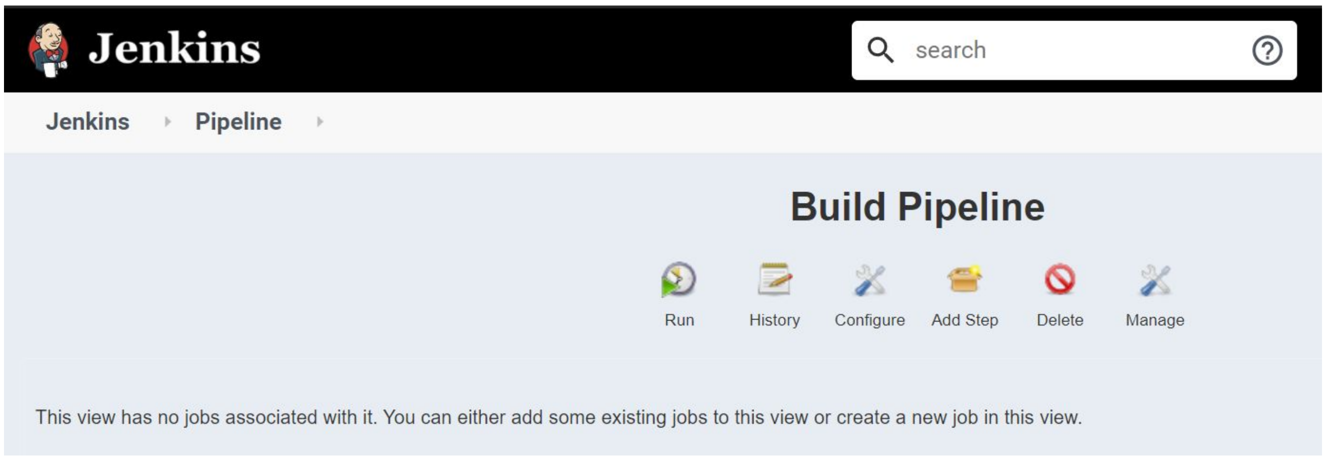
OMP is basically the command-line client of Open Vulnerability Assessment System (OpenVas) which interacts with the openvas manager using the omp protocol. The wrapper involves multiple steps:

- Create a target in the database using XML (A target can be defined only once)
- Create a Task while passing the unique id generated by the previous step as well as the type of scan (Full and very deep ultimate scan type is set to default)
- Start the scan by passing the Task ID to the omp binary.
- Generating a report from the scan ID (Multiple formats are supported)

Note: Task can be created any number of times which will be used to scan the server and again but target host can only be registered once.

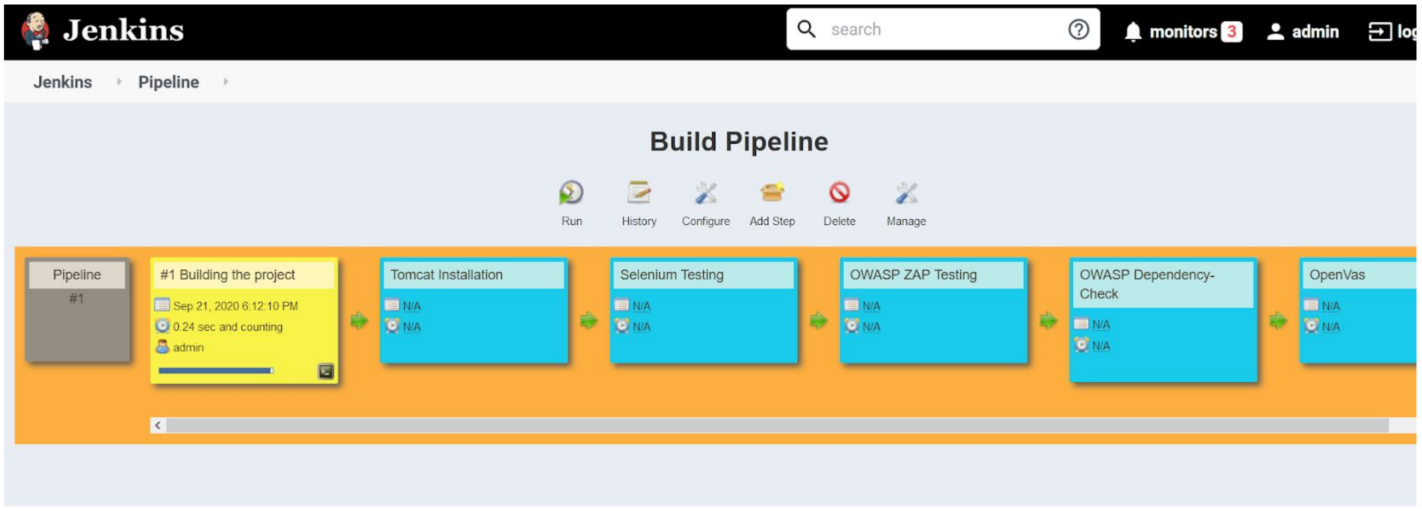
Pipeline Execution

Step 1: Navigate to the Pipeline tab.

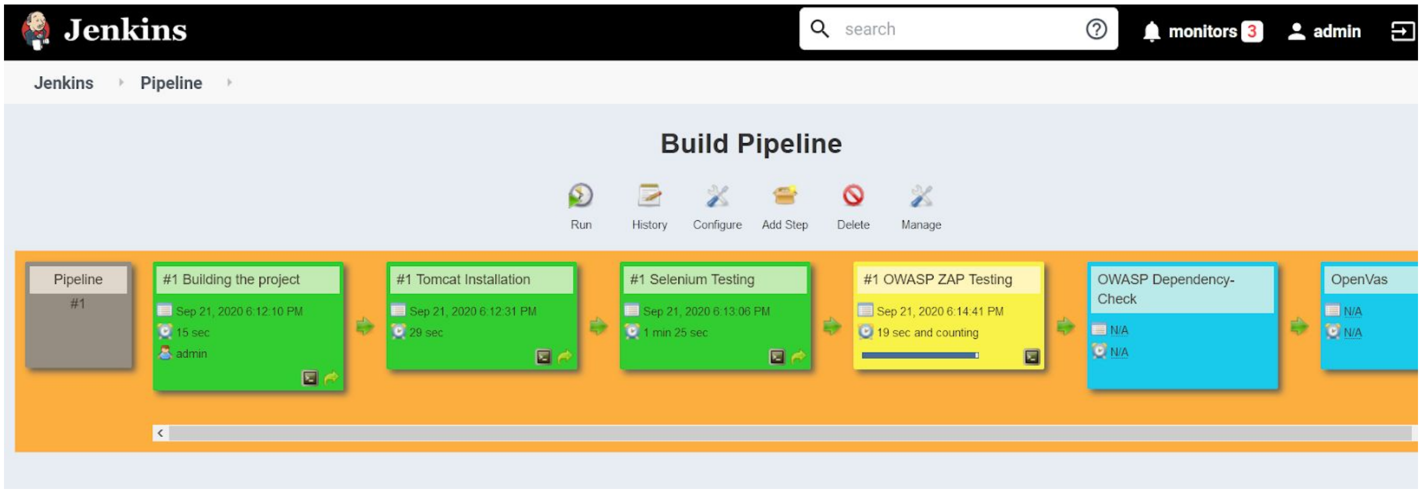


This view has no jobs associated with it. You can either add some existing jobs to this view or create a new job in this view.

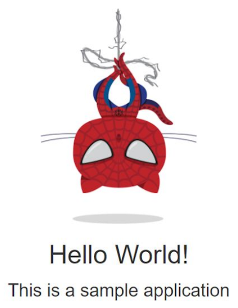
Step 2: Click on the Run button to start the Pipeline.



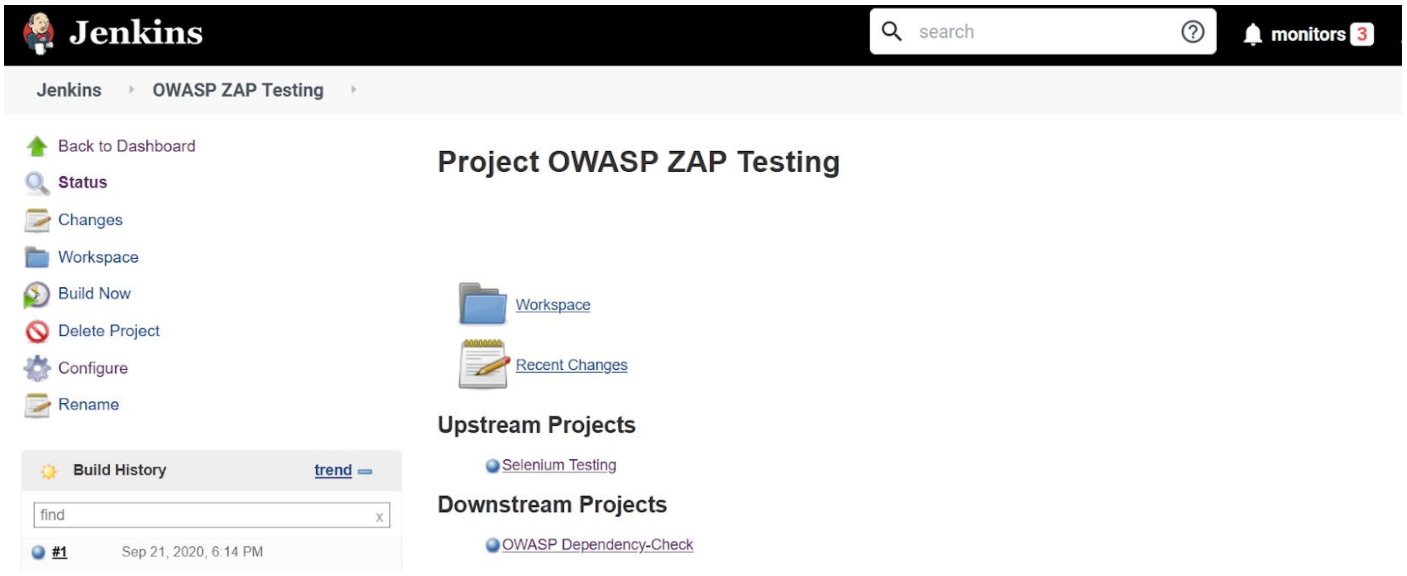
Reload the page to see the recent changes in the pipeline.



Once the pipeline is executed completely. The web application will be deployed on the test server. Check the test-server link to see the deployed application.



Step 3: Check the report generated by OWASP ZAP tool. Navigate to the OWASP ZAP Job page.


Jenkins | search | monitors 3

Jenkins › OWASP ZAP Testing ›

Back to Dashboard
Status
Changes
Workspace
Build Now
Delete Project
Configure
Rename

Build History trend

find

#1 Sep 21, 2020, 6:14 PM

Project OWASP ZAP Testing

Workspace

Recent Changes

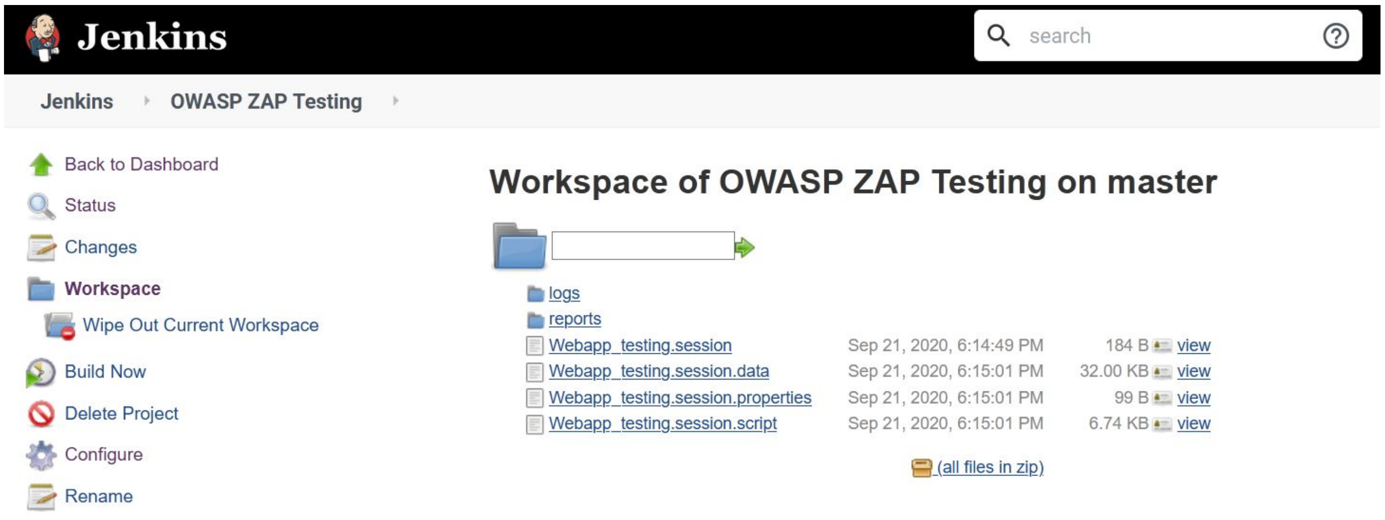
Upstream Projects

Selenium Testing

Downstream Projects

OWASP Dependency-Check

Click on the Workspace.


Jenkins | search

Jenkins › OWASP ZAP Testing ›

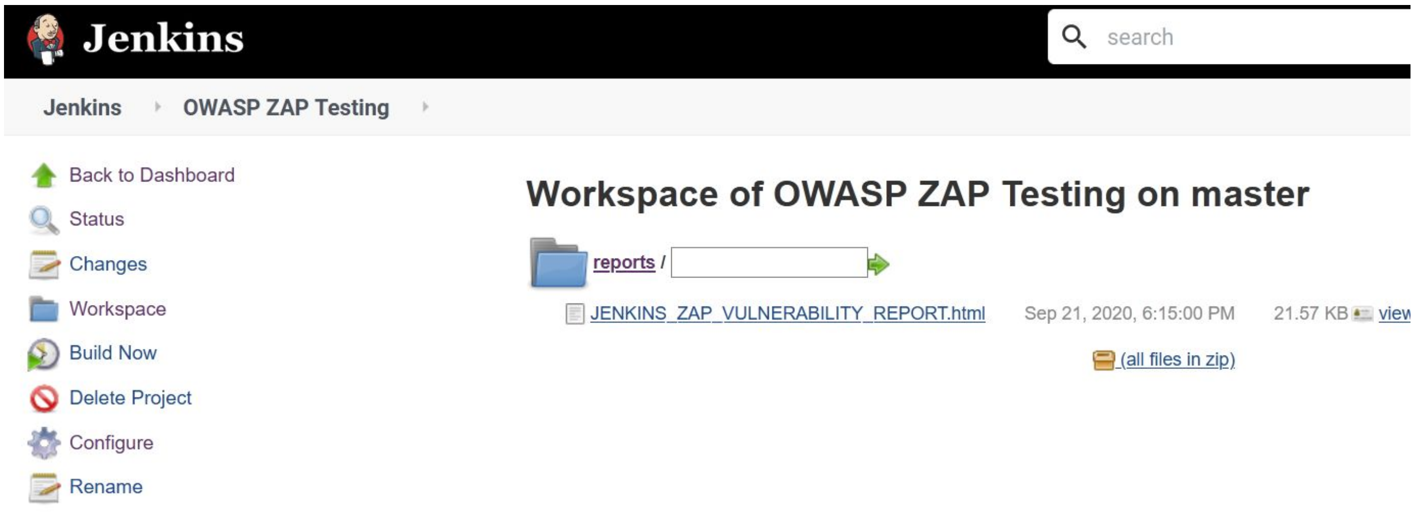
Back to Dashboard
Status
Changes
Workspace
Wipe Out Current Workspace
Build Now
Delete Project
Configure
Rename

Workspace of OWASP ZAP Testing on master

logs			
reports			
Webapp_testing.session	Sep 21, 2020, 6:14:49 PM	184 B	view
Webapp_testing.session.data	Sep 21, 2020, 6:15:01 PM	32.00 KB	view
Webapp_testing.session.properties	Sep 21, 2020, 6:15:01 PM	99 B	view
Webapp_testing.session.script	Sep 21, 2020, 6:15:01 PM	6.74 KB	view

(all files in zip)

Change to the reports directory.



Jenkins › OWASP ZAP Testing ›

- Back to Dashboard
- Status
- Changes
- Workspace
- Build Now
- Delete Project
- Configure
- Rename

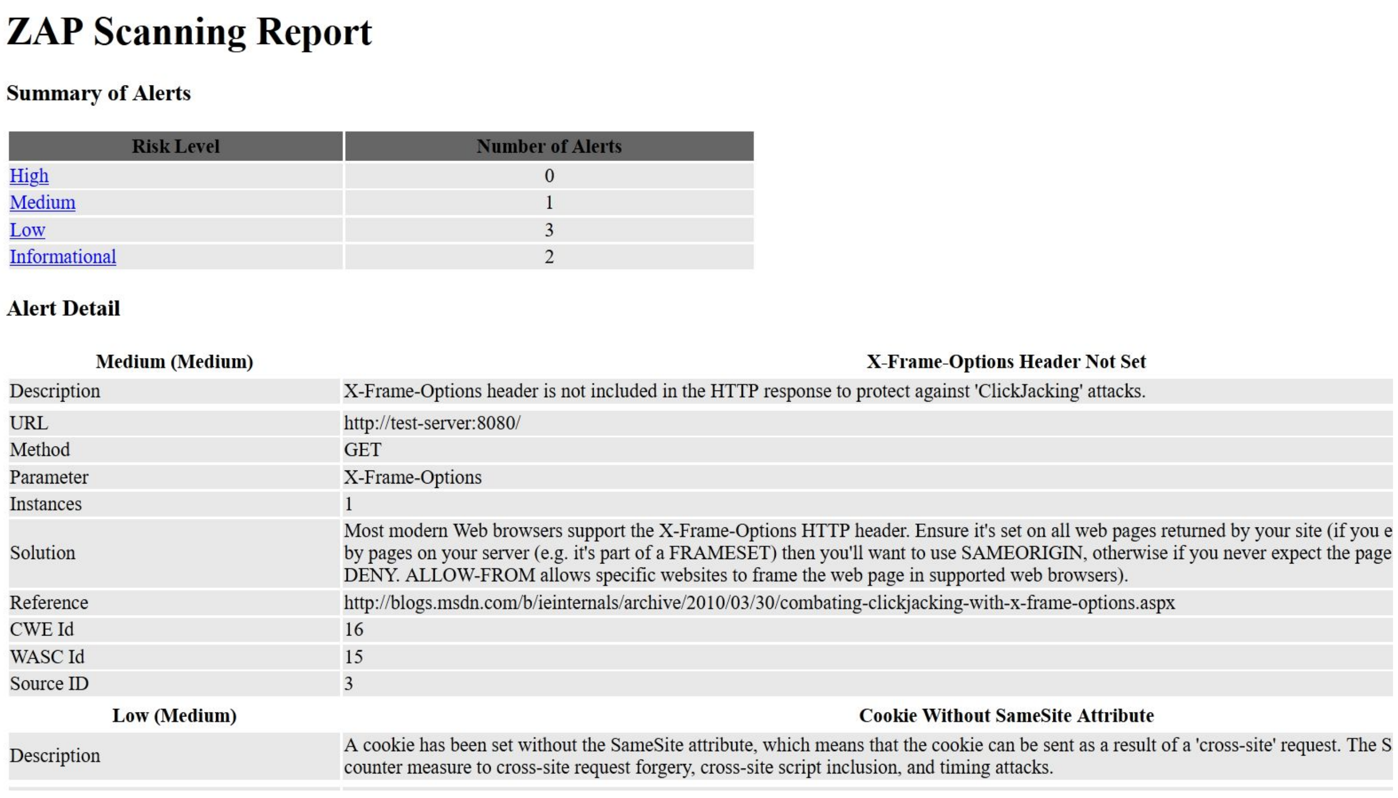
Workspace of OWASP ZAP Testing on master

reports /

JENKINS_ZAP_VULNERABILITY_REPORT.html	Sep 21, 2020, 6:15:00 PM	21.57 KB	view

(all files in zip)

Open the report file.



ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	3
Informational	2

Alert Detail

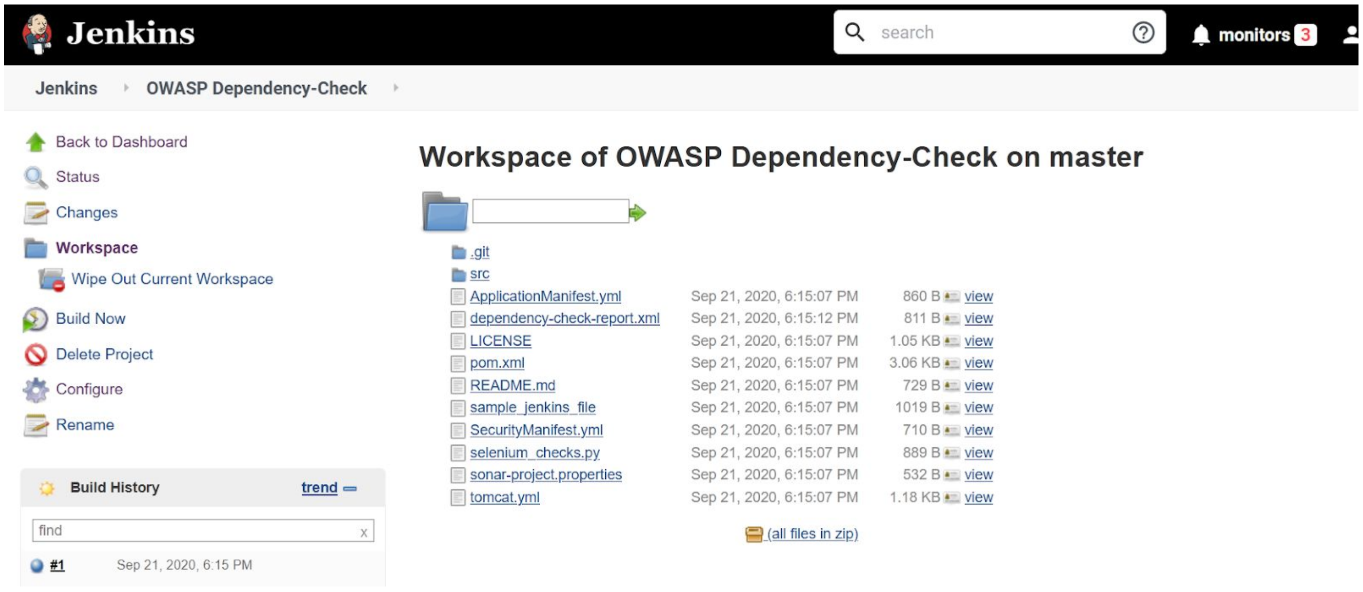
Medium (Medium)	X-Frame-Options Header Not Set
Description	X-Frame-Options header is not included in the HTTP response to protect against 'ClickJacking' attacks.
URL	http://test-server:8080/
Method	GET
Parameter	X-Frame-Options
Instances	1
Solution	Most modern Web browsers support the X-Frame-Options HTTP header. Ensure it's set on all web pages returned by your site (if you e by pages on your server (e.g. it's part of a FRAMESET) then you'll want to use SAMEORIGIN, otherwise if you never expect the page DENY. ALLOW-FROM allows specific websites to frame the web page in supported web browsers).
Reference	http://blogs.msdn.com/b/ieinternals/archive/2010/03/30/combating-clickjacking-with-x-frame-options.aspx
CWE Id	16
WASC Id	15
Source ID	3

Low (Medium)	Cookie Without SameSite Attribute
Description	A cookie has been set without the SameSite attribute, which means that the cookie can be sent as a result of a 'cross-site' request. The S counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Issues Detected:

- No X-Frame-Options header included which could lead to ClickJacking attacks
- Cookie without SameSite Attribute
- X-Content-Type-Options Header Missing
- Charset Mismatch (Header Versus Meta Charset)

Step 4: Check the XML report generated by OWASP Dependency Check which is located under the job's workspace

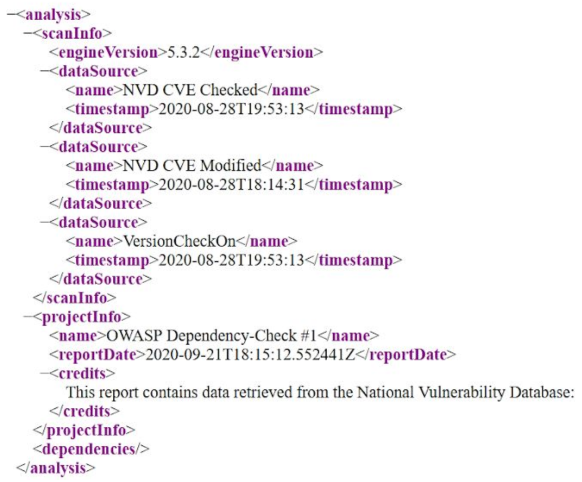


Workspace of OWASP Dependency-Check on master

.git			
src			
ApplicationManifest.yml	Sep 21, 2020, 6:15:07 PM	860 B	view
dependency-check-report.xml	Sep 21, 2020, 6:15:12 PM	811 B	view
LICENSE	Sep 21, 2020, 6:15:07 PM	1.05 KB	view
pom.xml	Sep 21, 2020, 6:15:07 PM	3.06 KB	view
README.md	Sep 21, 2020, 6:15:07 PM	729 B	view
sample_jenkins_file	Sep 21, 2020, 6:15:07 PM	1019 B	view
SecurityManifest.yml	Sep 21, 2020, 6:15:07 PM	710 B	view
selenium_checks.py	Sep 21, 2020, 6:15:07 PM	889 B	view
sonar-project.properties	Sep 21, 2020, 6:15:07 PM	532 B	view
tomcat.yml	Sep 21, 2020, 6:15:07 PM	1.18 KB	view

(all files in zip)

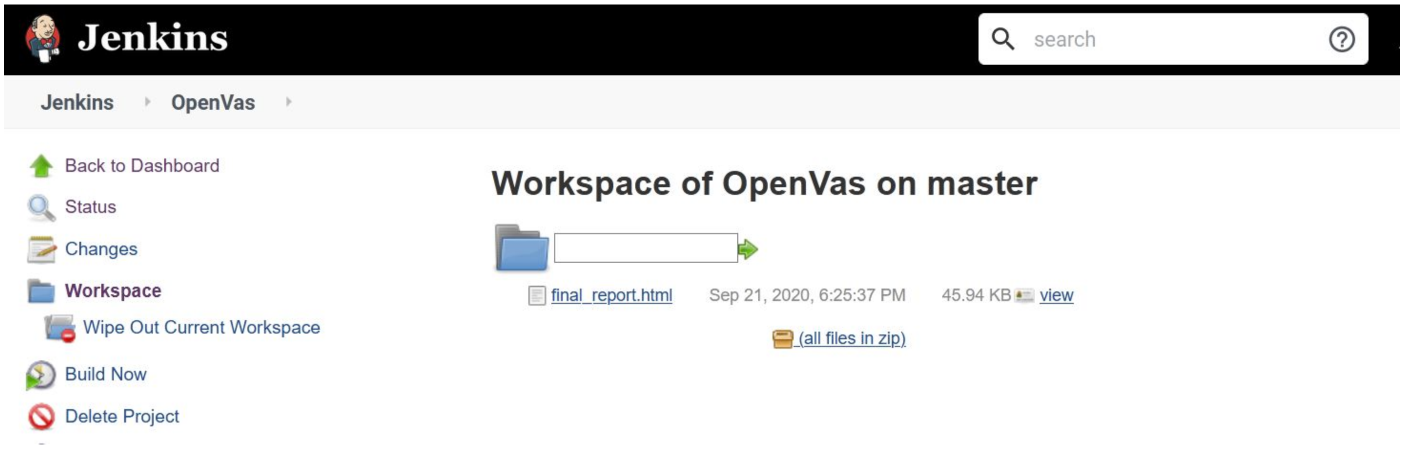
Open the "dependency-check-report.xml" file



```
–<analysis>
  –<scanInfo>
      <engineVersion>5.3.2</engineVersion>
    –<dataSource>
        <name>NVD CVE Checked</name>
        <timestamp>2020-08-28T19:53:13</timestamp>
      </dataSource>
    –<dataSource>
        <name>NVD CVE Modified</name>
        <timestamp>2020-08-28T18:14:31</timestamp>
      </dataSource>
    –<dataSource>
        <name>VersionCheckOn</name>
        <timestamp>2020-08-28T19:53:13</timestamp>
      </dataSource>
    </scanInfo>
  –<projectInfo>
      <name>OWASP Dependency-Check #1</name>
      <reportDate>2020-09-21T18:15:12.552441Z</reportDate>
    –<credits>
        This report contains data retrieved from the National Vulnerability Database:
      </credits>
    </projectInfo>
    <dependencies/>
  </analysis>
```

There were no dependency based issues in the code. Hence the dependency check report does not show any information.

Step 5: Check the report generated by OpenVas, The report is located in the workspace.

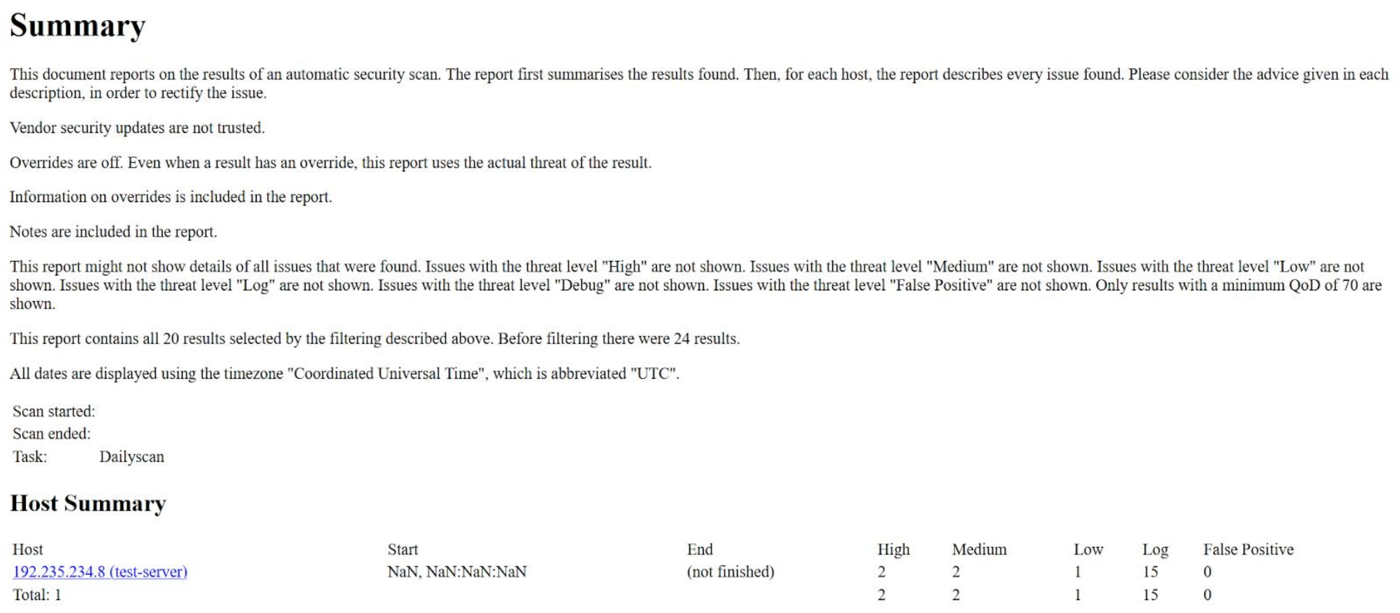


Workspace of OpenVas on master

final_report.html Sep 21, 2020, 6:25:37 PM 45.94 KB view

(all files in zip)

Open the html report.



Summary

This document reports on the results of an automatic security scan. The report first summarises the results found. Then, for each host, the report describes every issue found. Please consider the advice given in each description, in order to rectify the issue.

Vendor security updates are not trusted.

Overrides are off. Even when a result has an override, this report uses the actual threat of the result.

Information on overrides is included in the report.

Notes are included in the report.

This report might not show details of all issues that were found. Issues with the threat level "High" are not shown. Issues with the threat level "Medium" are not shown. Issues with the threat level "Low" are not shown. Issues with the threat level "Log" are not shown. Issues with the threat level "Debug" are not shown. Issues with the threat level "False Positive" are not shown. Only results with a minimum QoD of 70 are shown.

This report contains all 20 results selected by the filtering described above. Before filtering there were 24 results.

All dates are displayed using the timezone "Coordinated Universal Time", which is abbreviated "UTC".

Scan started:	
Scan ended:	
Task:	Dailyscan

Host Summary

Host	Start	End	High	Medium	Low	Log	False Positive
192.235.234.8 (test-server)	NaN, NaN:NaN:NaN	(not finished)	2	2	1	15	0
Total: 1			2	2	1	15	0

Issues Detected:

- Apache tomcat has default files installed
- Application transmits sensitive information via HTTP protocol
- HTTP Security Headers Detection
- Operating System Detection

Learning

Working of a simple DevSecOps pipeline consisting of different components.