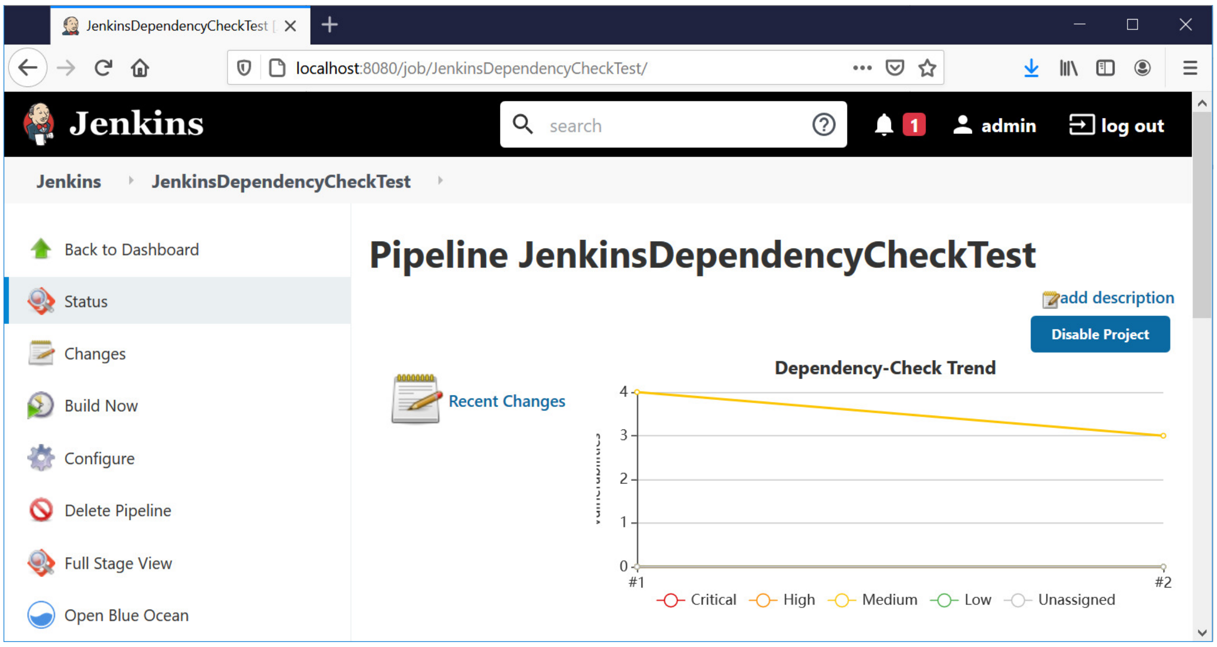
Integrating Jenkins with OWASP Dependency Check

Overview



In Lecture 6, you are introduced to the OWASP Top 10 Proactive Controls which describe the most important security controls that should be adopted in every secure software development project. One of the proactive controls is C2: Leverage Security Frameworks and Libraries, and one of the best practices for this control is to proactively keep them up to date possibly with the help of tools, e.g. OWASP Dependency Check – a Software Component Analysis (SCA) tool to detect publicly disclosed vulnerabilities contained within the dependencies of a project.

In this Lab 6, you are going to learn how to integrate OWASP Dependency Check into Jenkins Pipeline so that you are able to incorporate dependency check into your team project.

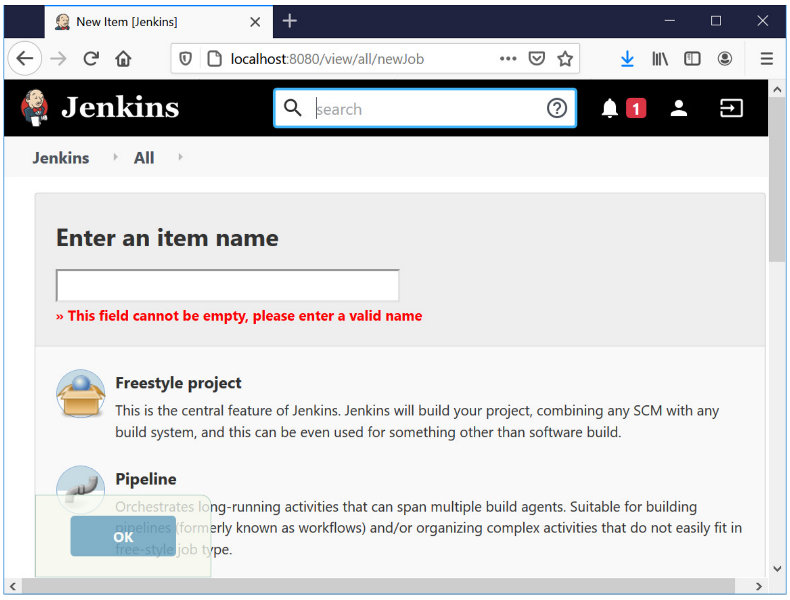
Outcomes

Upon completion of this session, you should be able to:

- Install OWASP Dependency Check plugin into Jenkins
- Create Jenkins pipeline to perform dependency check on a simple web application with vulnerable dependency
- Incorporate Jenkins Pipeline with suitable SCA, e.g. OWASP Dependency Check, into your team project

1: Reference

This lab is based on the blog at https://www.nagarrosecurity.com/blog/adding-owasp-dependency-check-to-jenkins. However, it was written using Jenkins Freestyle project which you may find it difficult to adapt it using the newer recommended Pipeline. (For an explanation of why Pipeline, please read https://www.jenkins.io/pipeline/getting-started-pipelines/.)



Enter an item name

» This field cannot be empty, please enter a valid name

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

To help you, this lab hand-out is re-written using Jenkins Pipeline so as to facilitate you to incorporate OWASP Dependency Check into your team project.

Advise: When trying Jenkins and dependency check for your team project, you are likely unable to find a reference in the Internet that can fully meet your needs. Nevertheless, it is hoped that through the team project, you are able to acquire the skill of adapting what was available, couple with your knowledge and understanding, and KEEP TRYING to eventually finding the solutions. This will be a valuable skill especially when you are going to join the industry soon after this trimester.

2: Pre-requisite

OWASP Dependency Check Plugin

Follow the instructions at https://www.nagarrosecurity.com/blog/adding-owasp-dependency-check-to-jenkins on installing the OWASP Dependency Check plugin.

3: Integrating OWASP Dependency Check into Jenkins Pipeline

Git

Clone a copy of the simple web app using an intentionally outdated jquery library with vulnerabilities from https://github.com/0xprime/JenkinsDependencyCheckTest into your local Git repository.

Create new Jenkins Pipeline

Create a new Pipeline for the cloned simple web app following the similar procedure as Lab-X05.

Jenkinsfile with Dependency Check

You will not be able to follow the instructions at https://www.nagarrosecurity.com/blog/adding-owasp-dependency-check-to-jenkins to 'Add build step' and 'Invoke Dependency Check' which is only meant for Freestyle project. Instead, copy the provided Jenkinsfile for this Lab-X06 from the LMS to the home directory of the simple web app in your local Git repository.

Commit it with the commands:

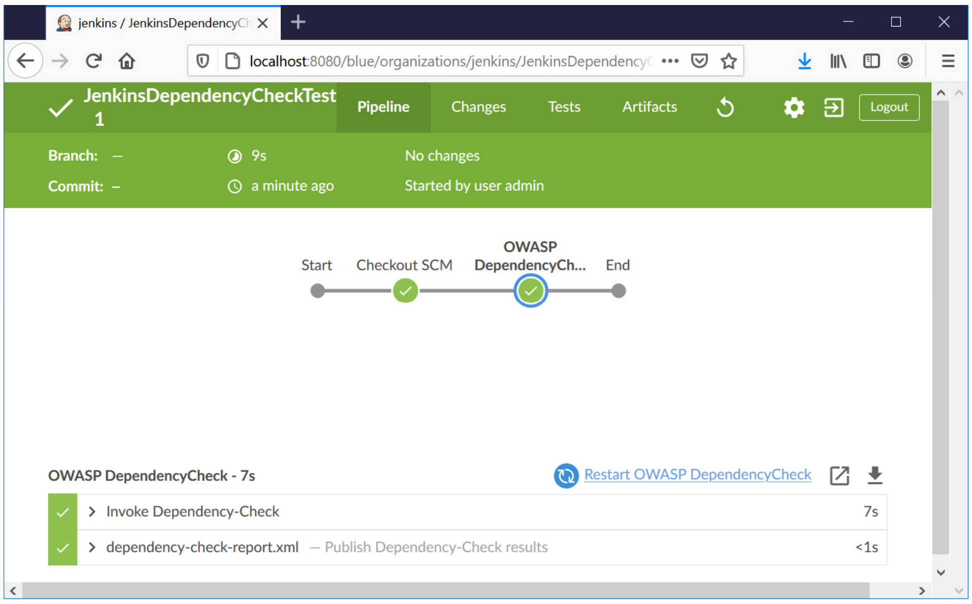
```
git add .
```

then

```
git commit -m "Add initial Jenkinsfile"
```

Run Jenkins Pipeline with Dependency Check

a) Run the Jenkins pipeline which should be successful as shown:

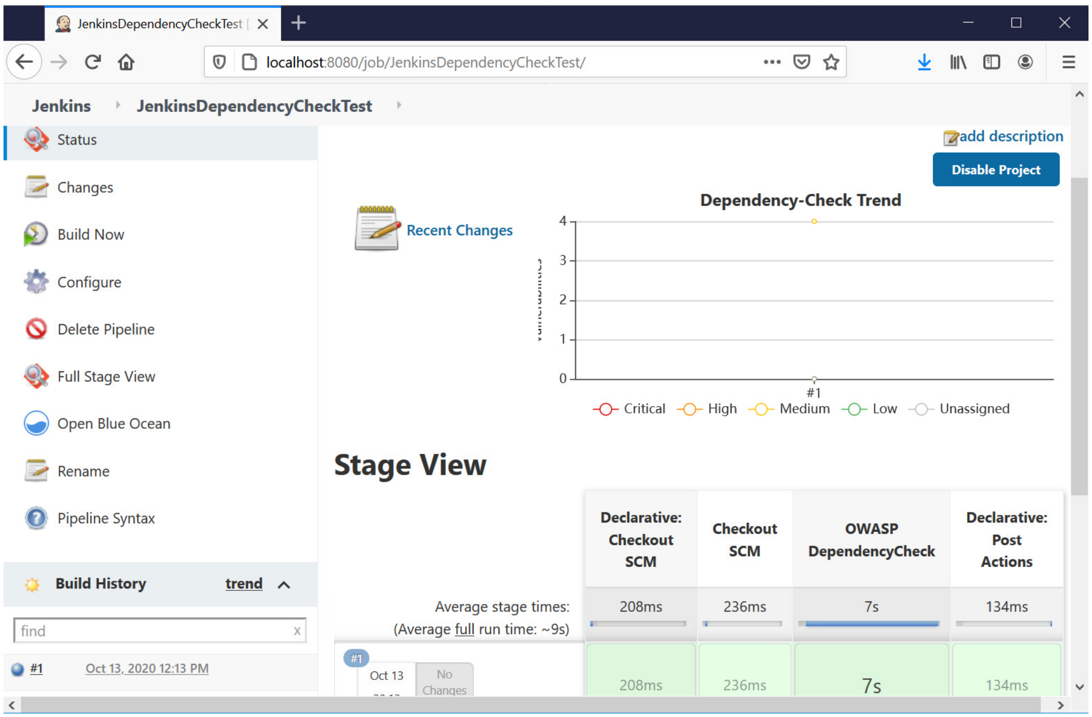

jenkins / JenkinsDependencyC
localhost:8080/blue/organizations/jenkins/JenkinsDependency
JenkinsDependencyCheckTest
1
Pipeline
Changes
Tests
Artifacts
Logout
Start
Checkout SCM
OWASP DependencyCh...
End
Restart OWASP DependencyCheck

Branch: –
Commit: –
9s
a minute ago
No changes
Started by user admin

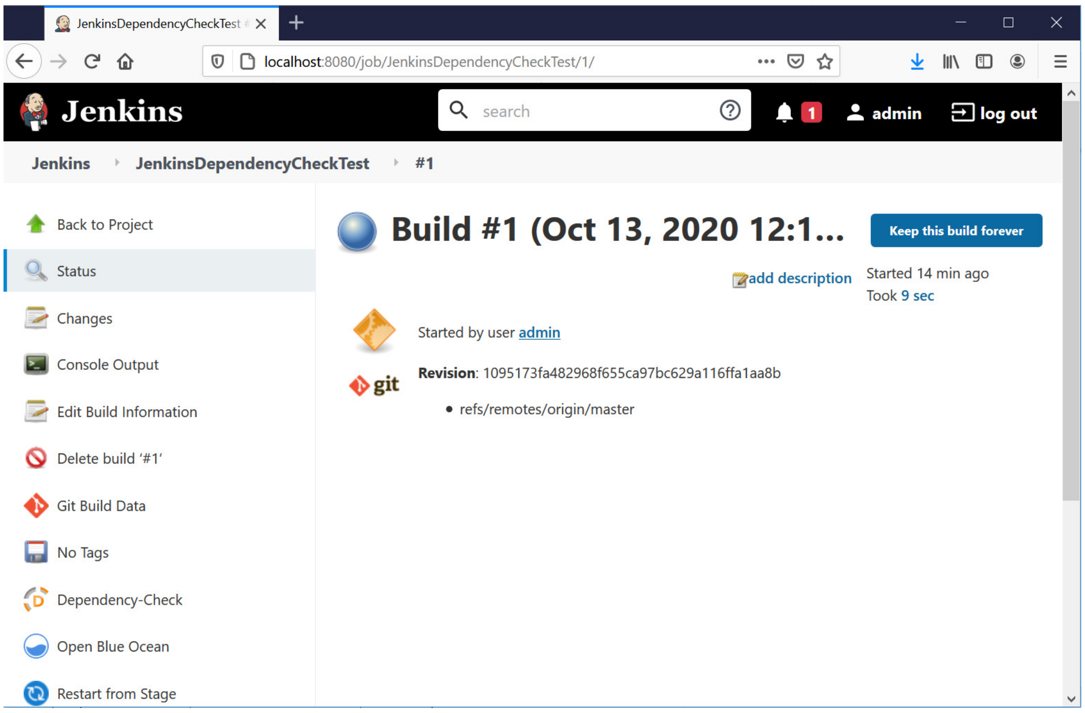
OWASP DependencyCheck - 7s

```
✓ > Invoke Dependency-Check                                                        7s
✓ > dependency-check-report.xml — Publish Dependency-Check results               <1s
```

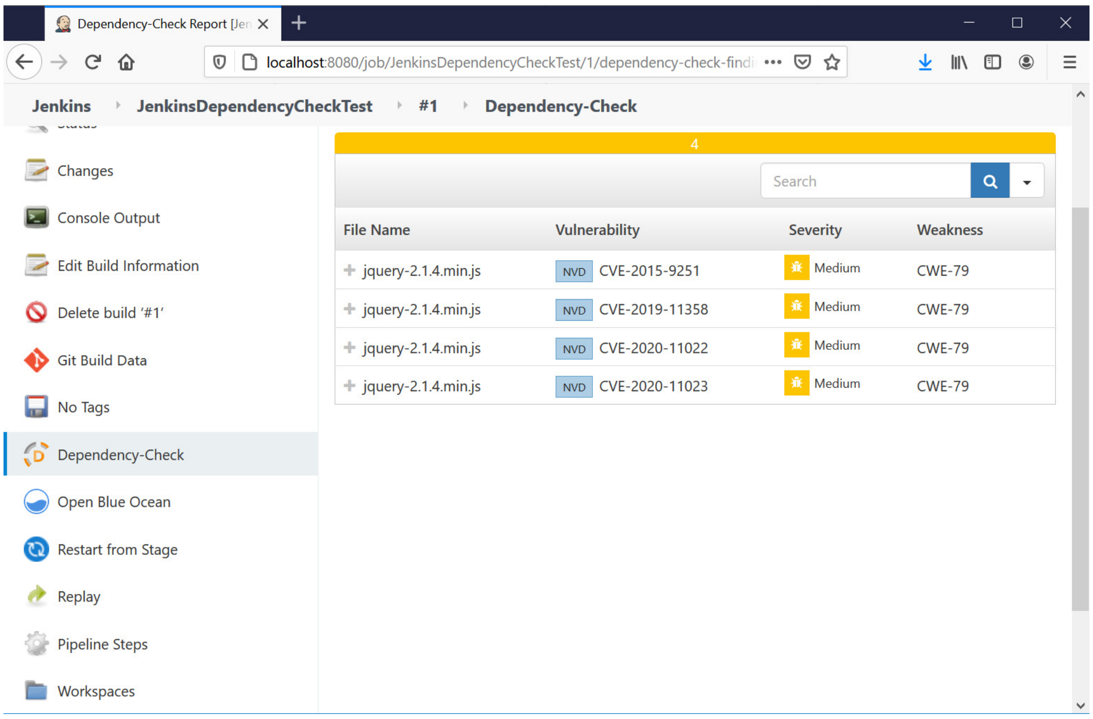
b) Exit from the Blue Ocean UI and return to Jenkins classic UI where you will see the Dependency-Check trend with 4 vulnerabilities (in yellow dot) reported as shown:



c) Click on #1 at the bottom left above which will bring you to Build #1 where you will see the 'Dependency-Check' icon on the left sidebar.

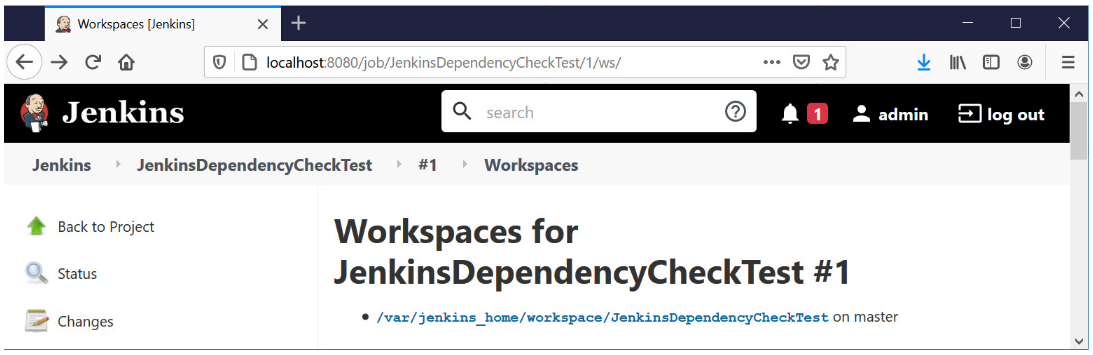


d) Click on the 'Dependency-Check' icon which will show you information of the 4 vulnerabilities found.



File Name	Vulnerability	Severity	Weakness
jquery-2.1.4.min.js	NVD CVE-2015-9251	Medium	CWE-79
jquery-2.1.4.min.js	NVD CVE-2019-11358	Medium	CWE-79
jquery-2.1.4.min.js	NVD CVE-2020-11022	Medium	CWE-79
jquery-2.1.4.min.js	NVD CVE-2020-11023	Medium	CWE-79

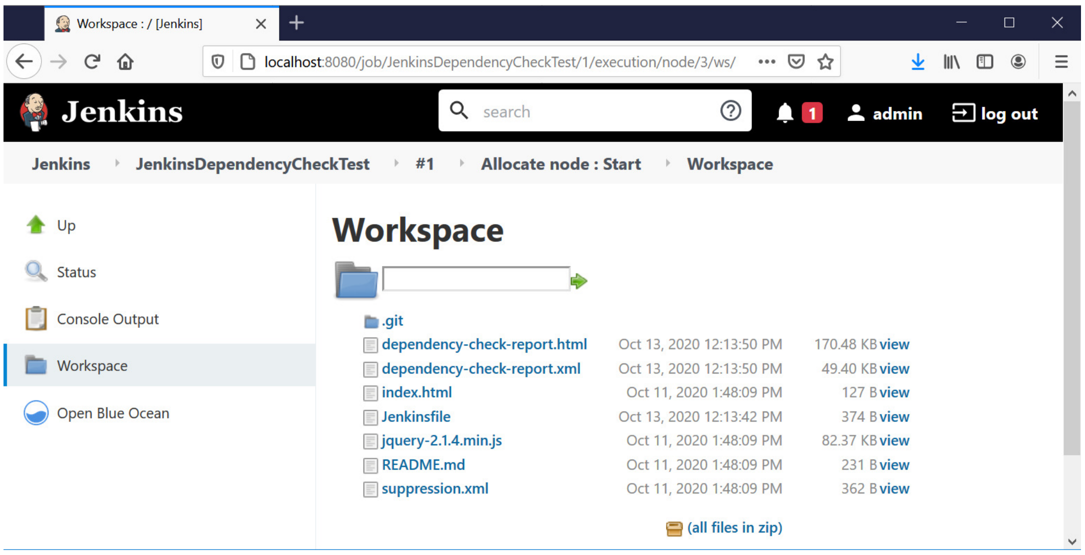
e) Click on the 'Workspaces' icon at the bottom left above which will bring you to the workspace for Build #1.


Workspaces [Jenkins]
localhost:8080/job/JenkinsDependencyCheckTest/1/ws/
Jenkins
search
1
admin
log out
Jenkins › JenkinsDependencyCheckTest › #1 › Workspaces
Back to Project
Status
Changes

Workspaces for JenkinsDependencyCheckTest #1

- `/var/jenkins_home/workspace/JenkinsDependencyCheckTest` on master

f) Click on the link in the workspace above which will show you the files for Build #1 as shown:



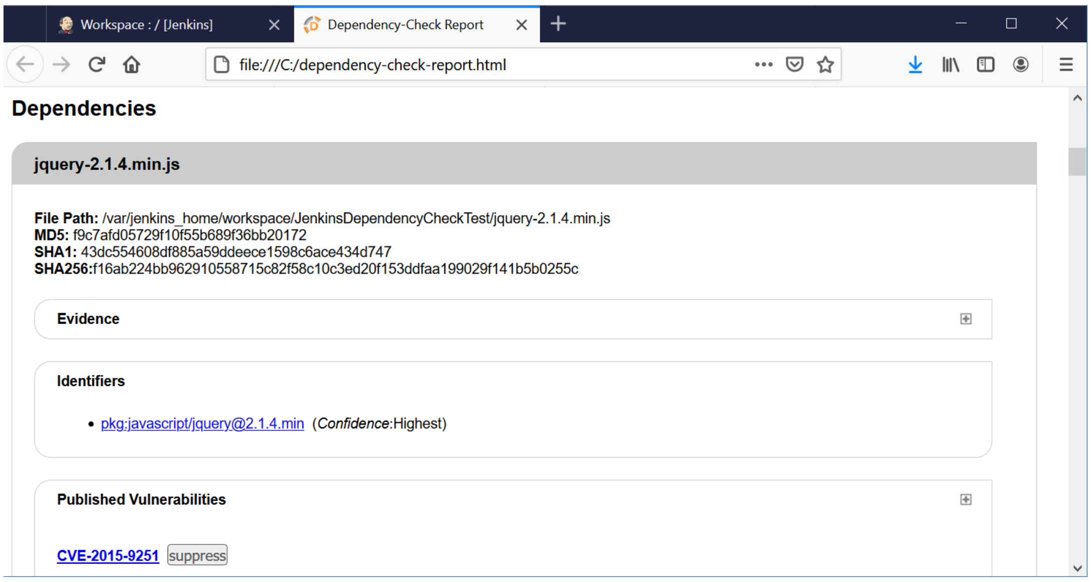
Workspace : / [Jenkins]
localhost:8080/job/JenkinsDependencyCheckTest/1/execution/node/3/ws/
Jenkins
search
admin
log out
Jenkins › JenkinsDependencyCheckTest › #1 › Allocate node : Start › Workspace
Up
Status
Console Output
Workspace
Open Blue Ocean

Workspace

.git			
dependency-check-report.html	Oct 13, 2020 12:13:50 PM	170.48 KB	view
dependency-check-report.xml	Oct 13, 2020 12:13:50 PM	49.40 KB	view
index.html	Oct 11, 2020 1:48:09 PM	127 B	view
Jenkinsfile	Oct 13, 2020 12:13:42 PM	374 B	view
jquery-2.1.4.min.js	Oct 11, 2020 1:48:09 PM	82.37 KB	view
README.md	Oct 11, 2020 1:48:09 PM	231 B	view
suppression.xml	Oct 11, 2020 1:48:09 PM	362 B	view

(all files in zip)

g) Due to the security of Jenkins, the dependency-check-report.html file will not be correctly displayed when viewed directly on the Jenkins server. Instead, download it locally and then open with a web browser which will show you the details of the vulnerabilities found.



Dependencies

jquery-2.1.4.min.js

File Path: /var/jenkins_home/workspace/JenkinsDependencyCheckTest/jquery-2.1.4.min.js
MD5: f9c7afd05729f10f55b689f36bb20172
SHA1: 43dc554608df885a59ddeece1598c6ace434d747
SHA256:f16ab224bb962910558715c82f58c10c3ed20f153ddfaa199029f141b5b0255c

Evidence

Identifiers

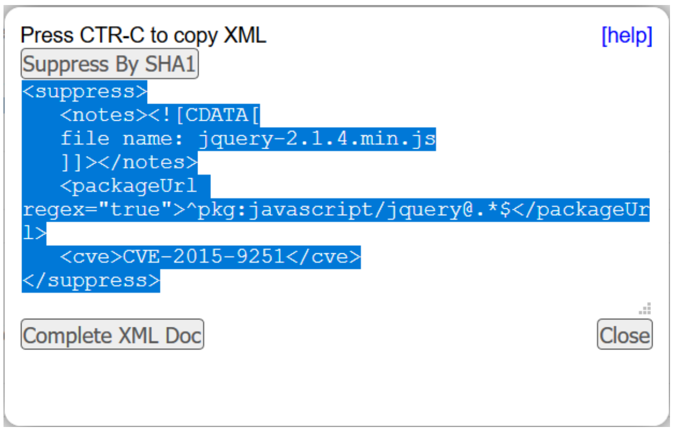
- pkg:javascript/jquery@2.1.4.min (*Confidence*:Highest)

Published Vulnerabilities

CVE-2015-9251 suppress

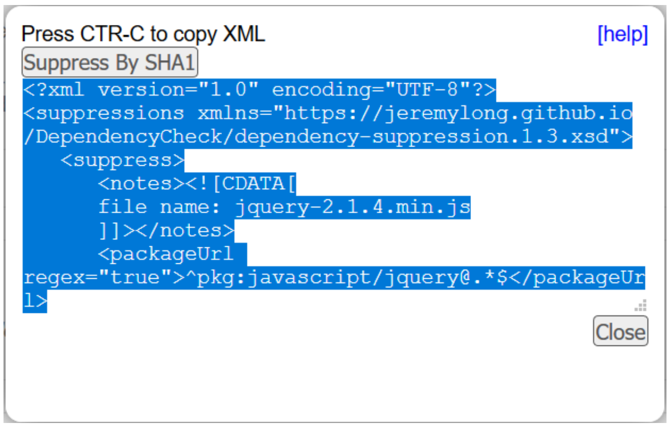
Dealing with False Positive

h) Suppose that there is a false positive that you wish to suppress, say CVE-2015-9251 as shown above. Click the 'suppress' button which will bring up a pop-up box:



```
<suppress>
   <notes><![CDATA[
   file name: jquery-2.1.4.min.js
   ]]></notes>
   <packageUrl
regex="true">^pkg:javascript/jquery@.*$</packageUrl>
   <cve>CVE-2015-9251</cve>
</suppress>
```

i) Click on the ‘Complete XML Doc’ button and then press CTR-C to copy the XML into a file and name it, say suppression.xml, and save it in the same location as the Jenkinsfile.



```
<?xml version="1.0" encoding="UTF-8"?>
<suppressions xmlns="https://jeremylong.github.io/DependencyCheck/dependency-suppression.1.3.xsd">
   <suppress>
      <notes><![CDATA[
      file name: jquery-2.1.4.min.js
      ]]></notes>
      <packageUrl
regex="true">^pkg:javascript/jquery@.*$</packageUrl>
```

j) Next, update the stage(‘OWASP DependencyCheck’) in the Jenkinsfile to include suppression.xml as shown:

```
steps {
    dependencyCheck additionalArguments: '--format HTML --format XML --suppression suppression.xml', odcInstallation: 'Default'
}
```

k) Commit the change and re-run the Jenkins pipeline which will result in the number of vulnerabilities being reduced to 3 for Build #2 as shown on the Dependency-Check Trend graph on the first page of this hand-out.

Congratulations! You have successfully completed the integration of OWASP Dependency Check into Jenkins pipeline. Hope that you are now ready to try incorporating dependency check into your team project. Enjoy!

END OF DOCUMENT