
ATTACK
DEFENSE
by PentesterAcademy

| Name | Defect Dojo: Managing Vulnerabilities |
|---|---|
| URL | https://attackdefense.com/challengedetails?cid=2272 |
| Type | DevSecOps Basics: Vulnerability Management |

**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

Defect Dojo is a security management tool for managing the application security program. The findings of different security tools can be collected using Defect Dojo. It helps in better tracking and patching of the vulnerabilities and other security issues.

A Kali machine is provided in this lab. There is also an instance of Defect-dojo running which can be accessed from inside the kali machine.

The Defect-dojo instance is accessible on hostname 'defect-dojo' and uses the following credentials:

| Username | Password |
|---|---|
| admin | admin |

The user is provided with access to Kali GUI and Defect-dojo web UI.

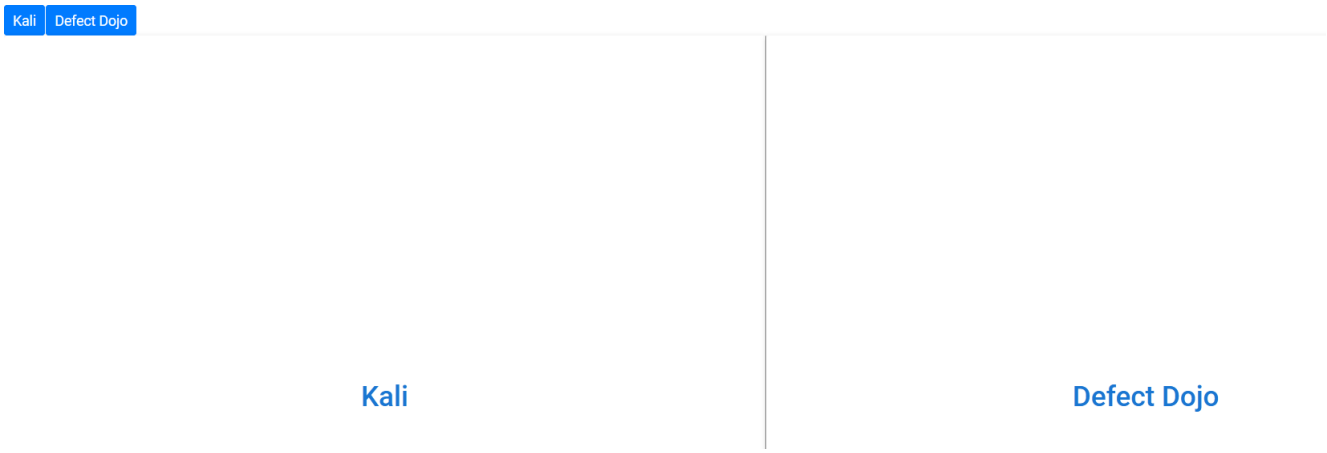
**Objective:** Follow the manual and learn how to use the Defect-dojo tool!

**Instructions:**

- The Defect Dojo instance is accessible with the name "defect-dojo" at port 8000.
- The source code of applications is provided at /root/github-repos

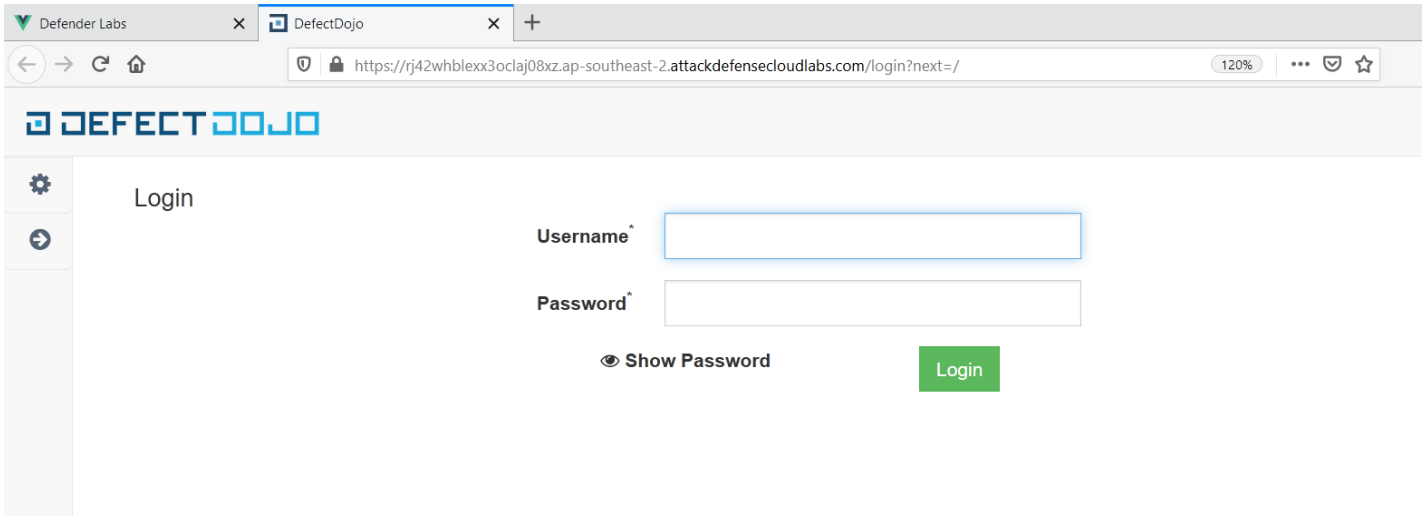
## Lab Setup

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




On choosing (clicking the text in the center) left panel, **a Kali GUI instance** will open in a new tab.

Similarly on selecting the right panel, a web UI of **Defect-dojo** will open in a new tab.

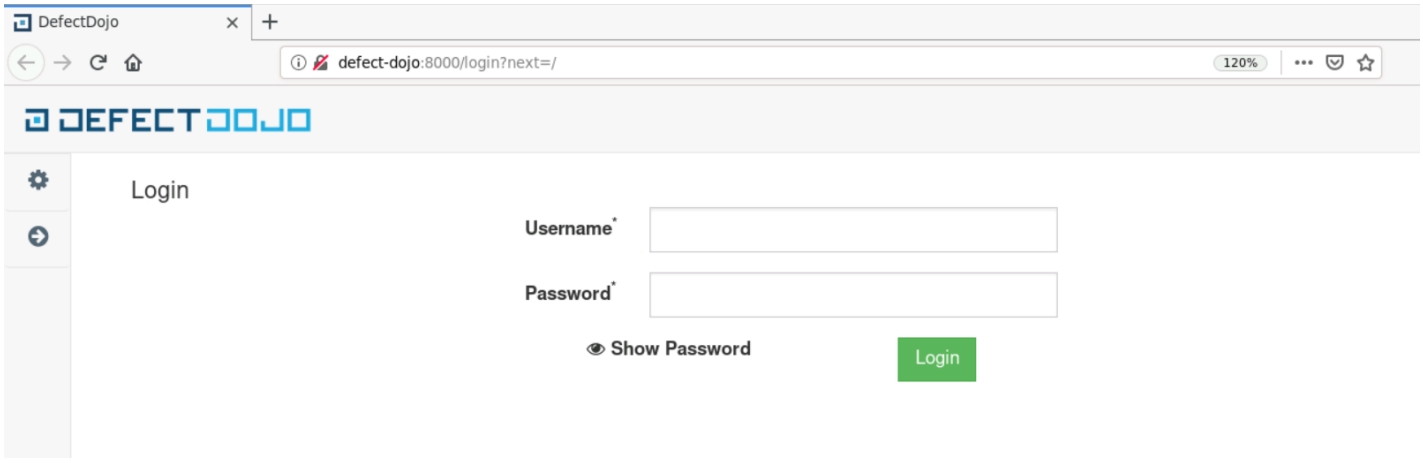



Now, both interfaces are ready and the user can start with the lab.

## Solution

**Step 1:** Open firefox in the kali instance and navigate to the defect-dojo website which can be reached at port 8000.

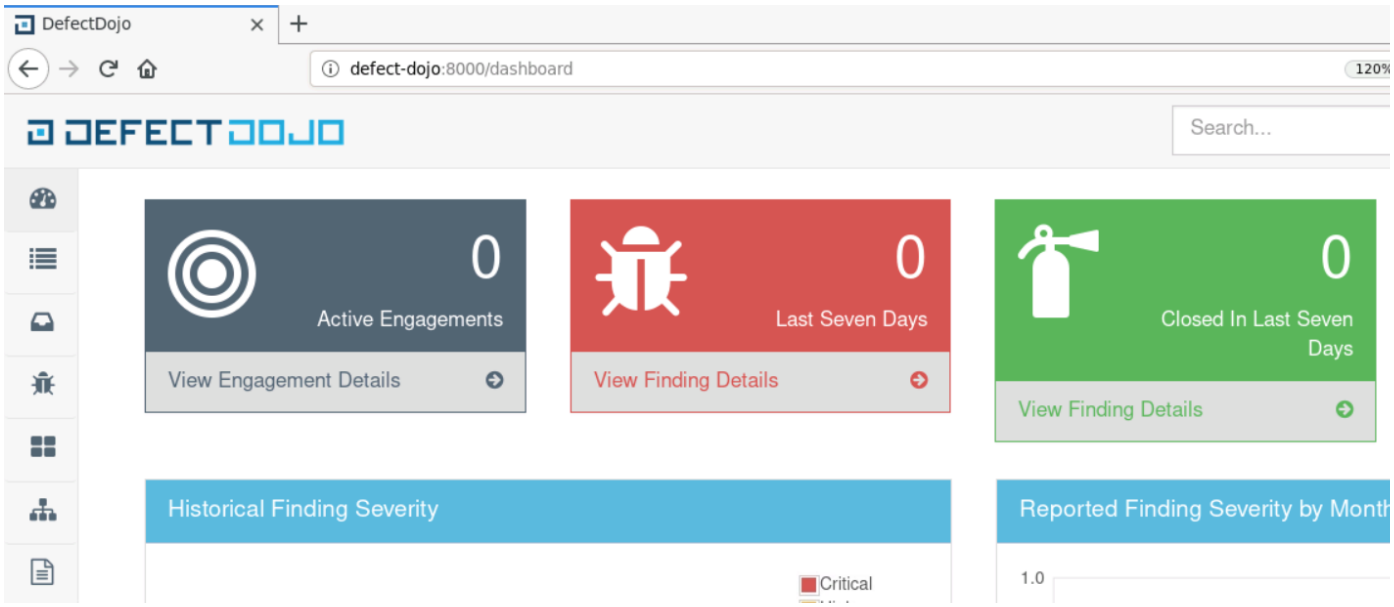
**URL:** http://defect-dojo:8000



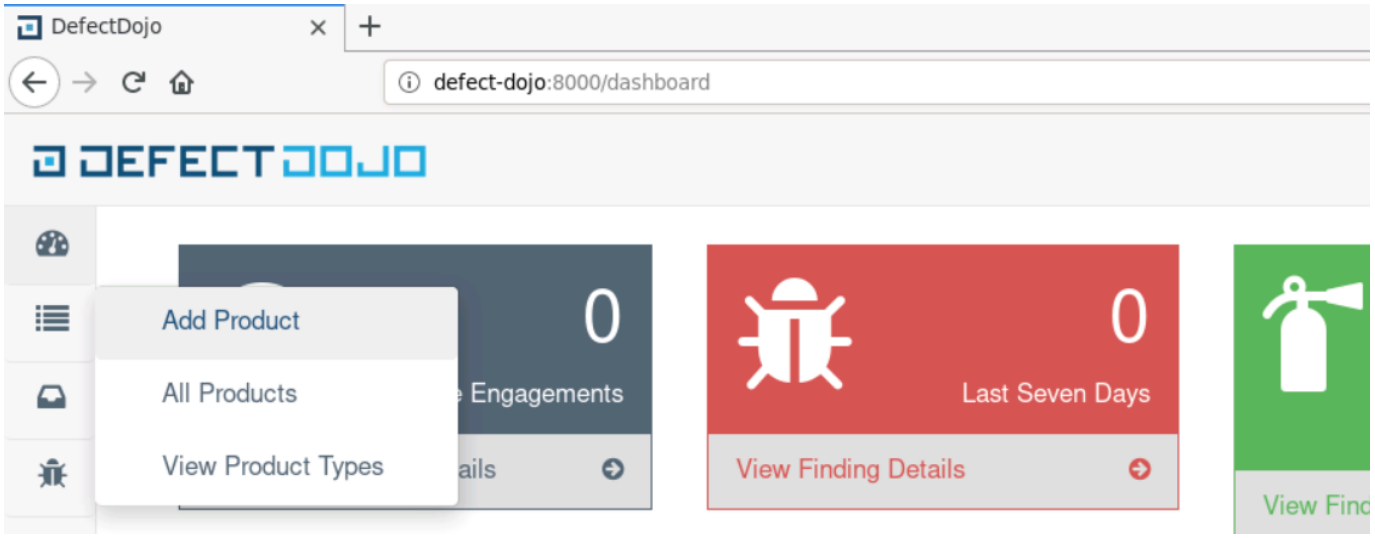
**Step 2:** Login using the credentials provided in the challenge description.

**Credentials:**

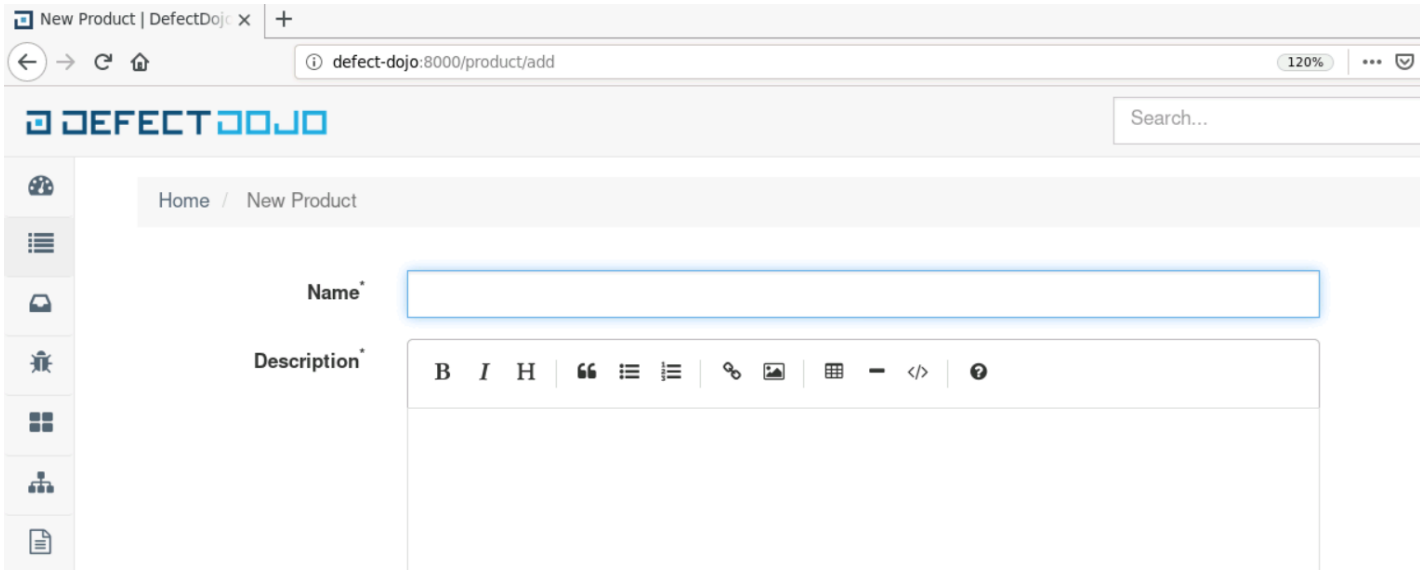
- **Username:** admin
- **Password:** admin



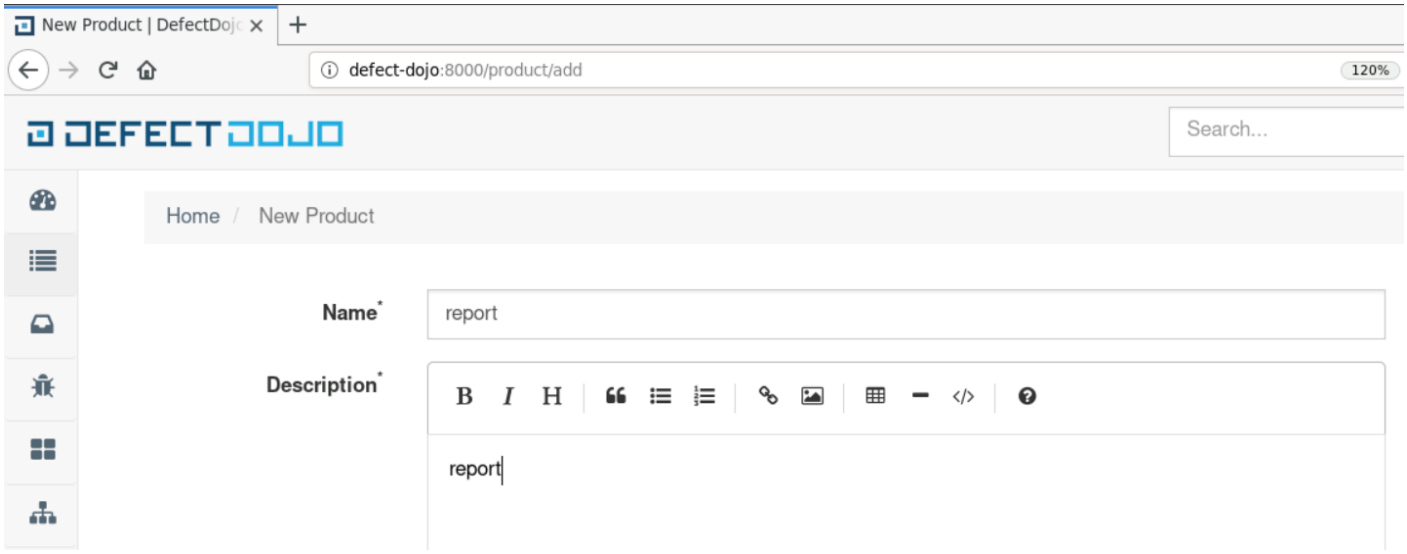

**Step 3:** Choose the “Add Product” from the product category



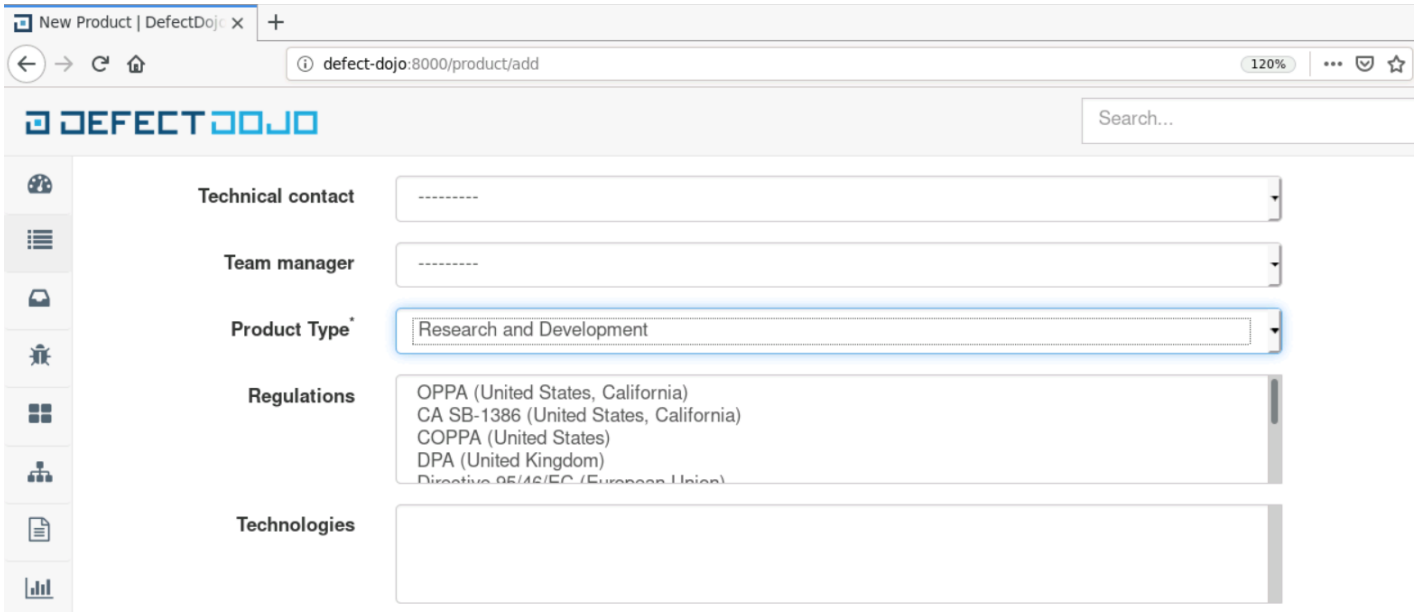
By clicking on the Add product, another page will open up.



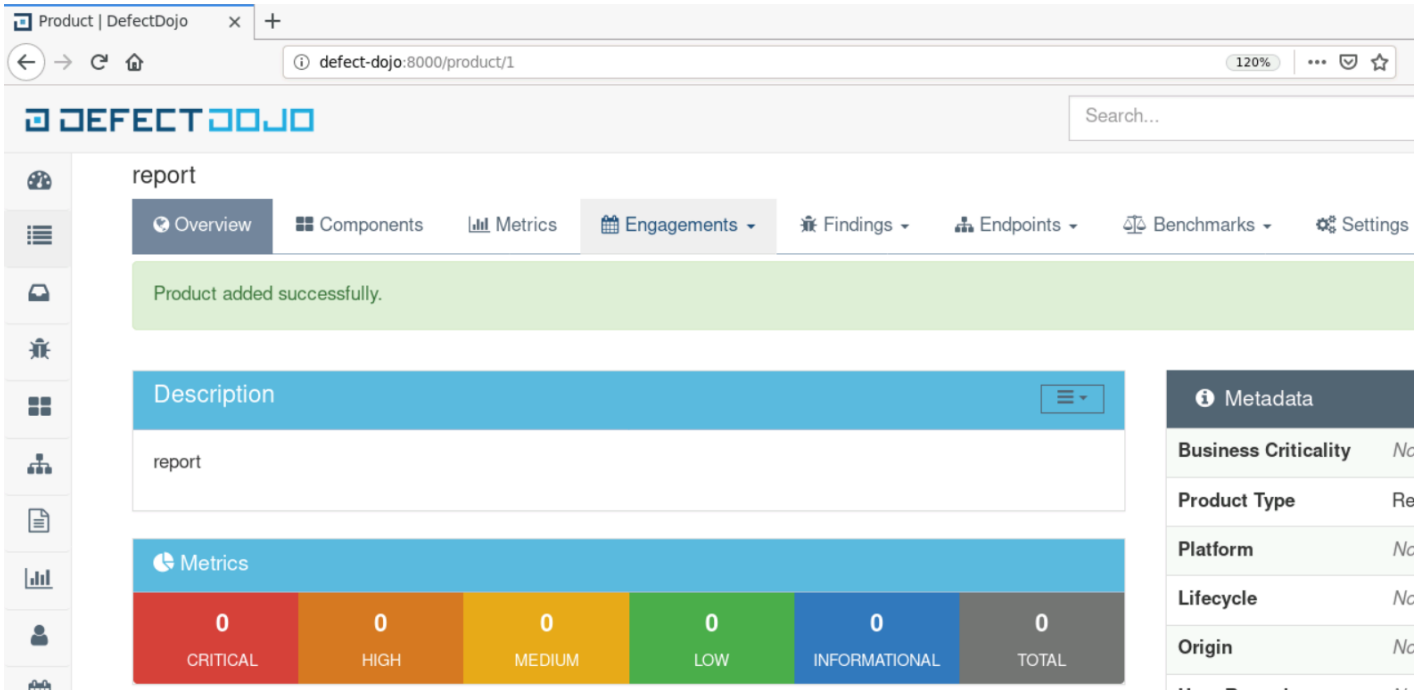

**Step 4:** Add any product name and enter any description.



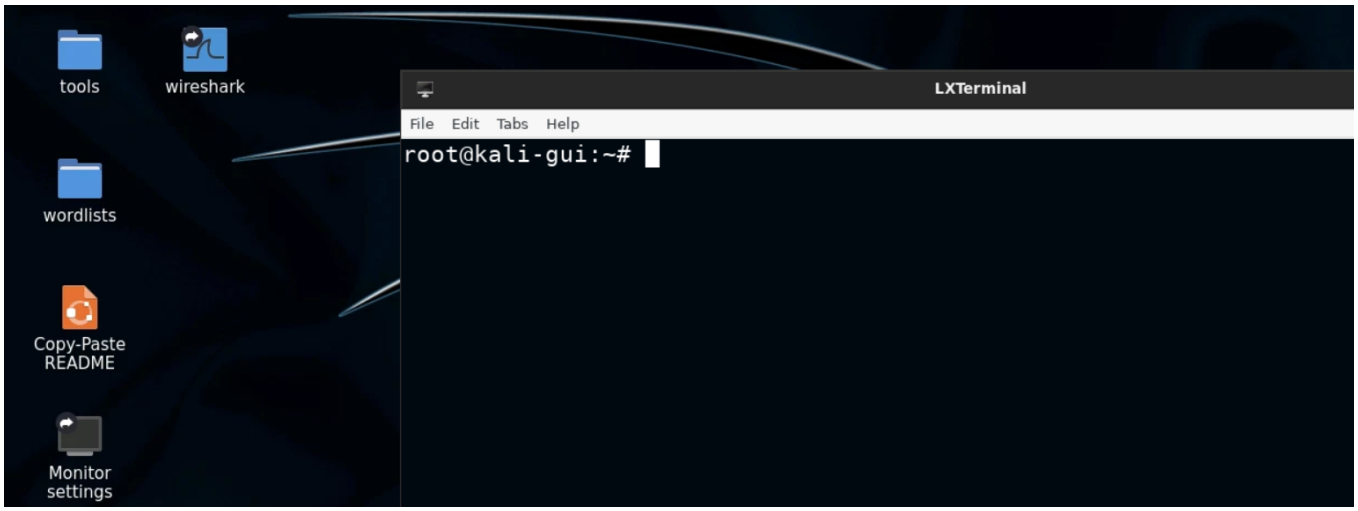

Scroll down to the ‘Product Type’ section and choose the ‘Research and Development’ option.




Scroll down and click on the “Submit” button.




**Step 5:** Start a terminal




**Step 6:** Check the provided applications.

**Command:** ls -l github-repos

```
root@kali-gui:~# ls -l github-repos/
total 8
drwxr-xr-x 4 root root 4096 Nov 20 16:57 is-online
drwxr-xr-x 7 root root 4096 Nov 20 16:57 node-app-template
root@kali-gui:~#
```

We will take one example at a time and run the tool on that.

It is important to understand that vulnerability management doesn't really perform any scan on their own but collect/correlate the issues/vulnerabilities flagged by other tools.

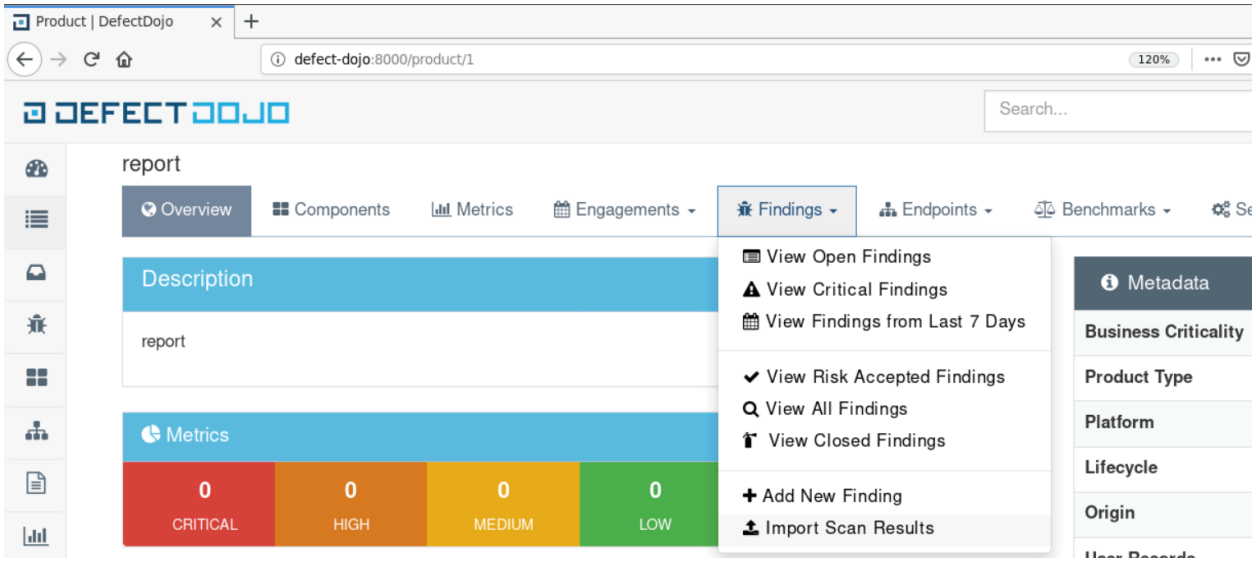
**Example 1: is-online**

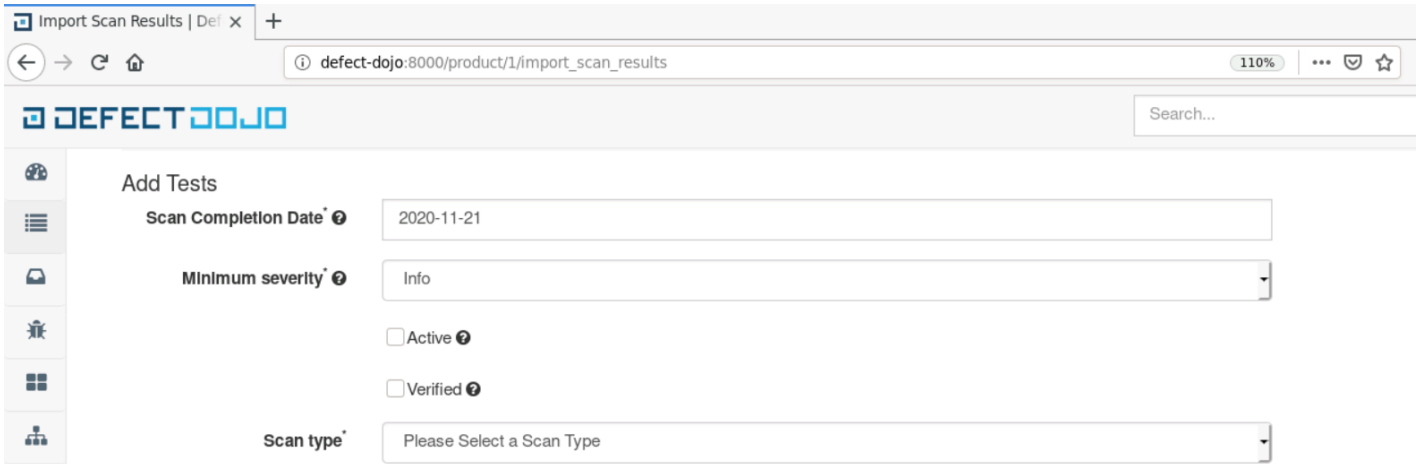
**Step 1:** Change to the is-online directory and run the Retire.js tool.

**Commands:**
cd github-repos/is-online
retire --js --outputformat json  --outputpath output.json
ls

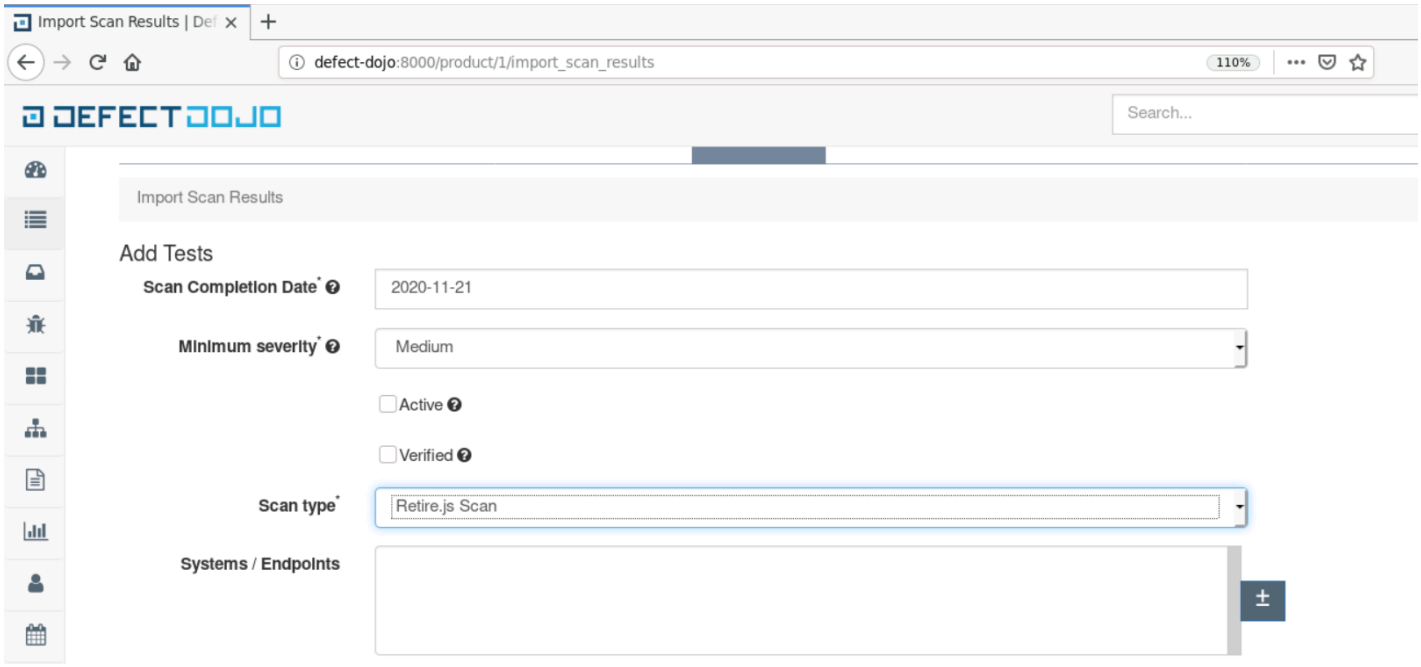
```
root@kali-gui:~# cd github-repos/is-online
root@kali-gui:~/github-repos/is-online# retire --js --outputformat json  --outputpath output.json
root@kali-gui:~/github-repos/is-online# ls
README.pa   dojo-release-1.3.3  index.js         license       output.json   readme.md        test.js
browser.js  index.d.ts          index.test-d.ts  node_modules  package.json  test-browser.js
root@kali-gui:~/github-repos/is-online#
```

**Step 2:** Navigate back to firefox and click on the “Import Scan Results” located under the ‘Findings’ section

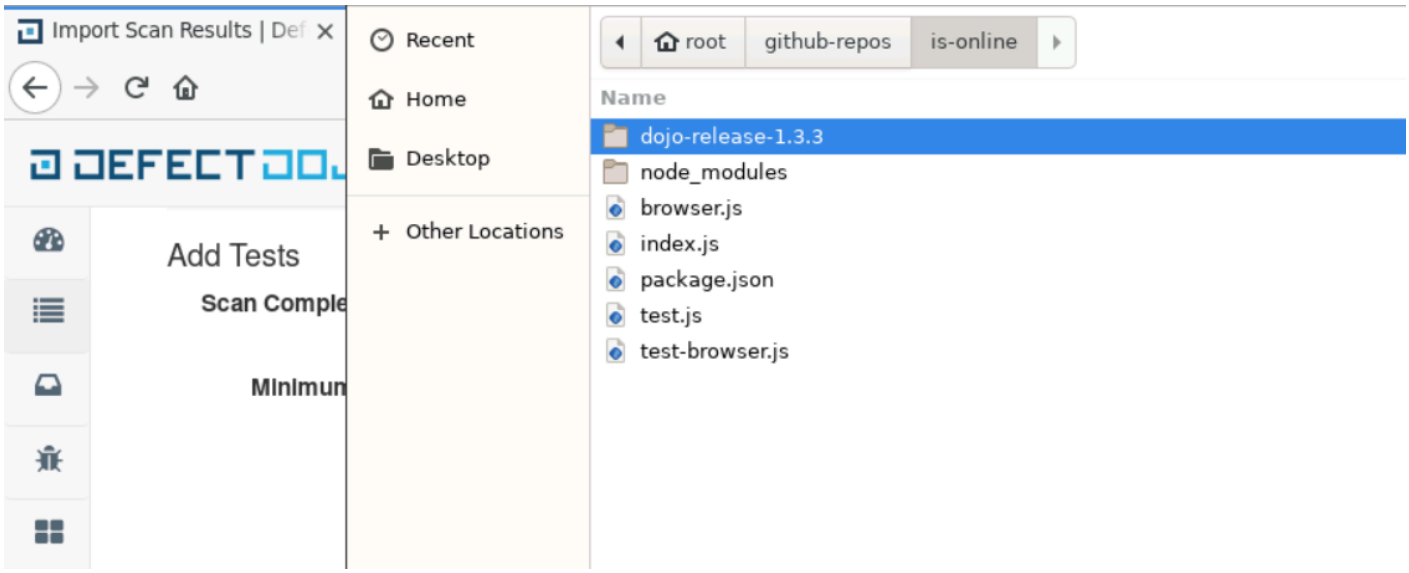


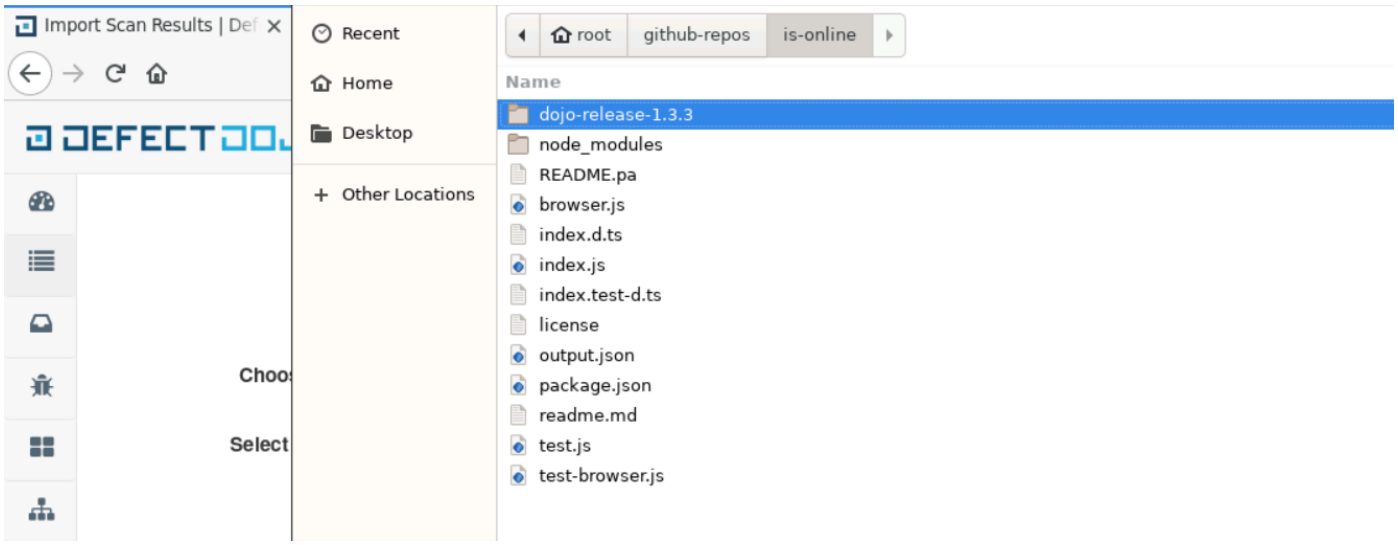
**Step 3:** Set the severity to “medium” as well as select “Retire.js Scan” in the ‘Scan Type’ section.



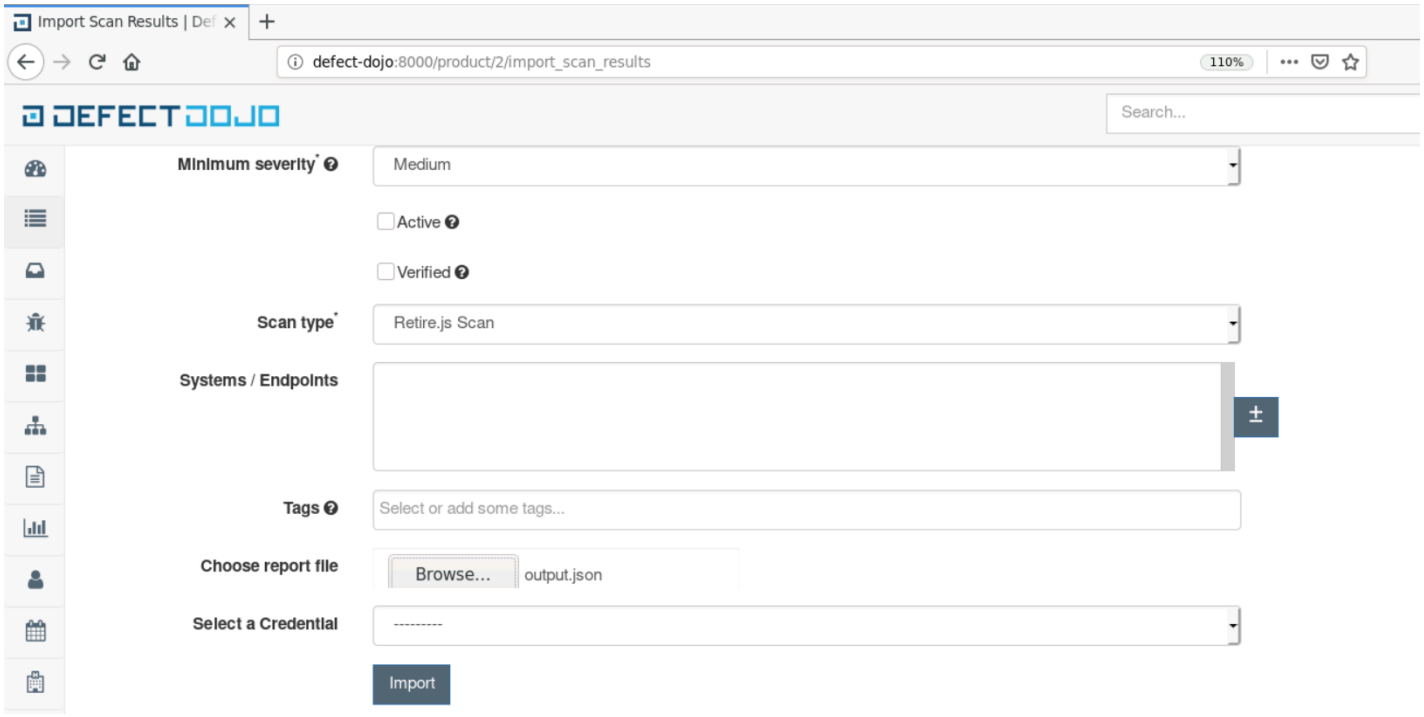

**Step 4:** Click on the ‘Choose report file’’s upload button and upload the ‘output.json’ file generated by retire js tool.



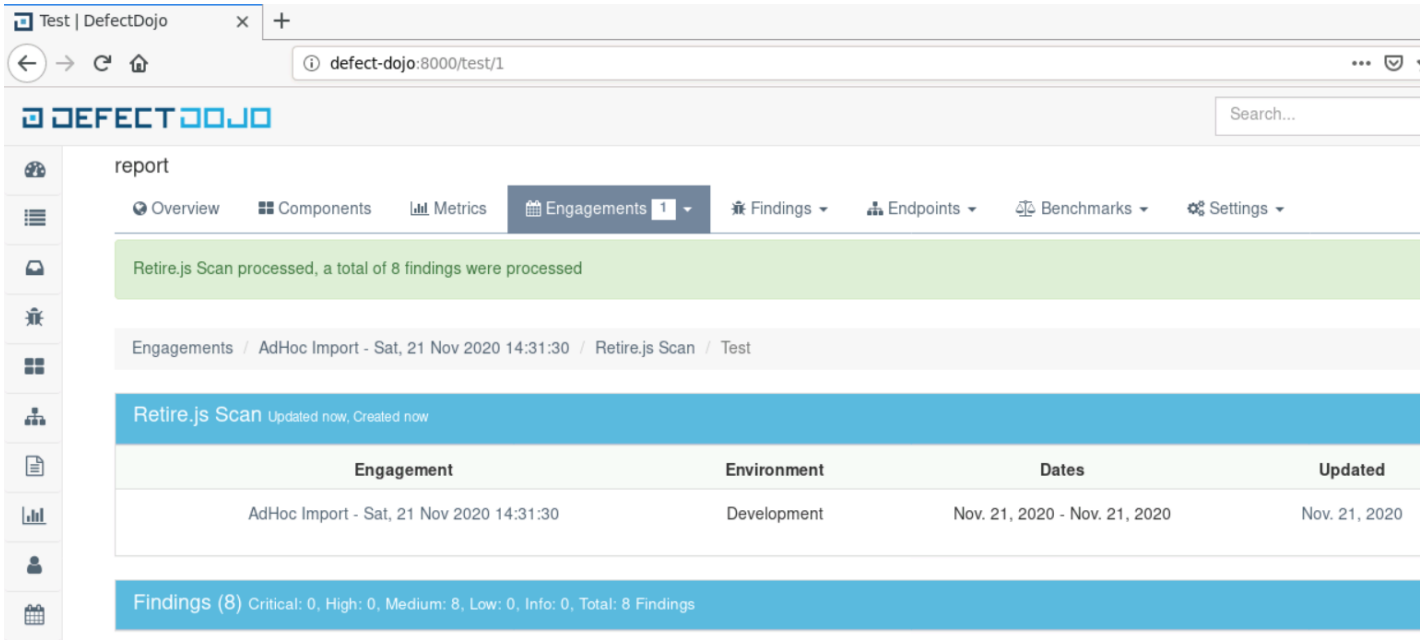
In the ‘is-online’ directory the file does not appear, select the ‘All files’ from the drop-down menu located in the down right of the dialogue box.



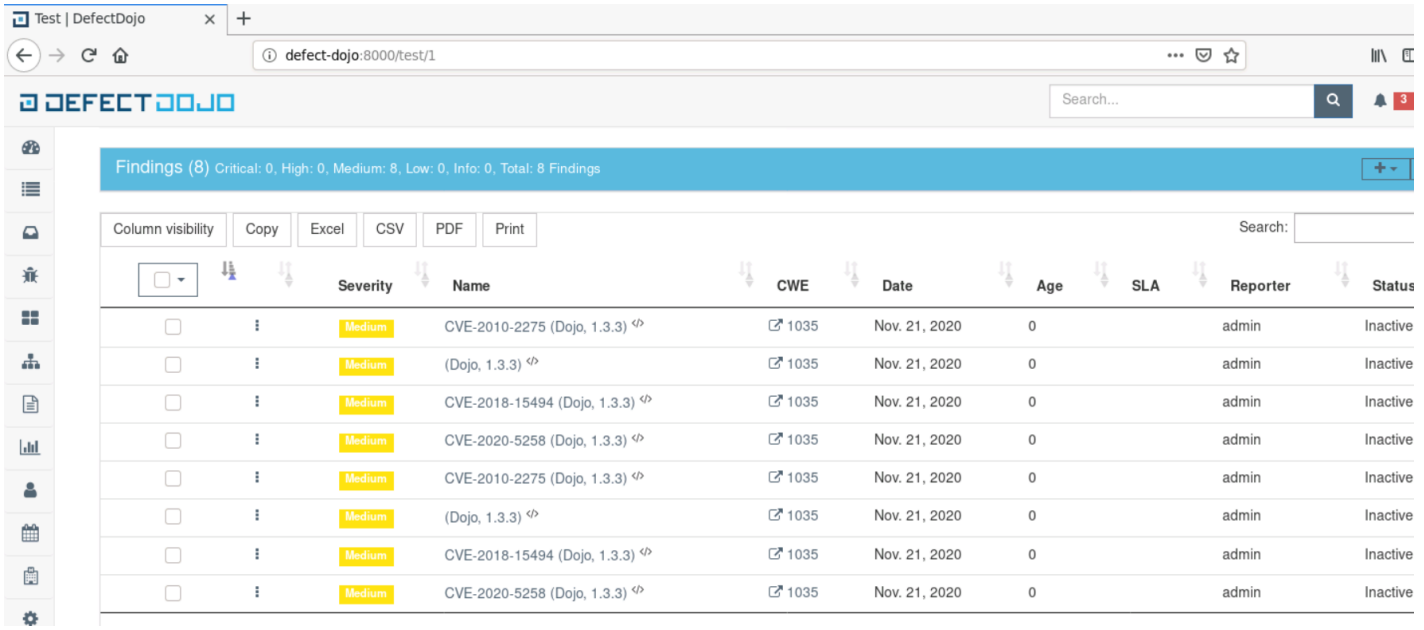

Select the ‘Output.json’ file and click on the Open button.



**Step 5:** Click on the Import button.




Scroll down to check the vulnerabilities

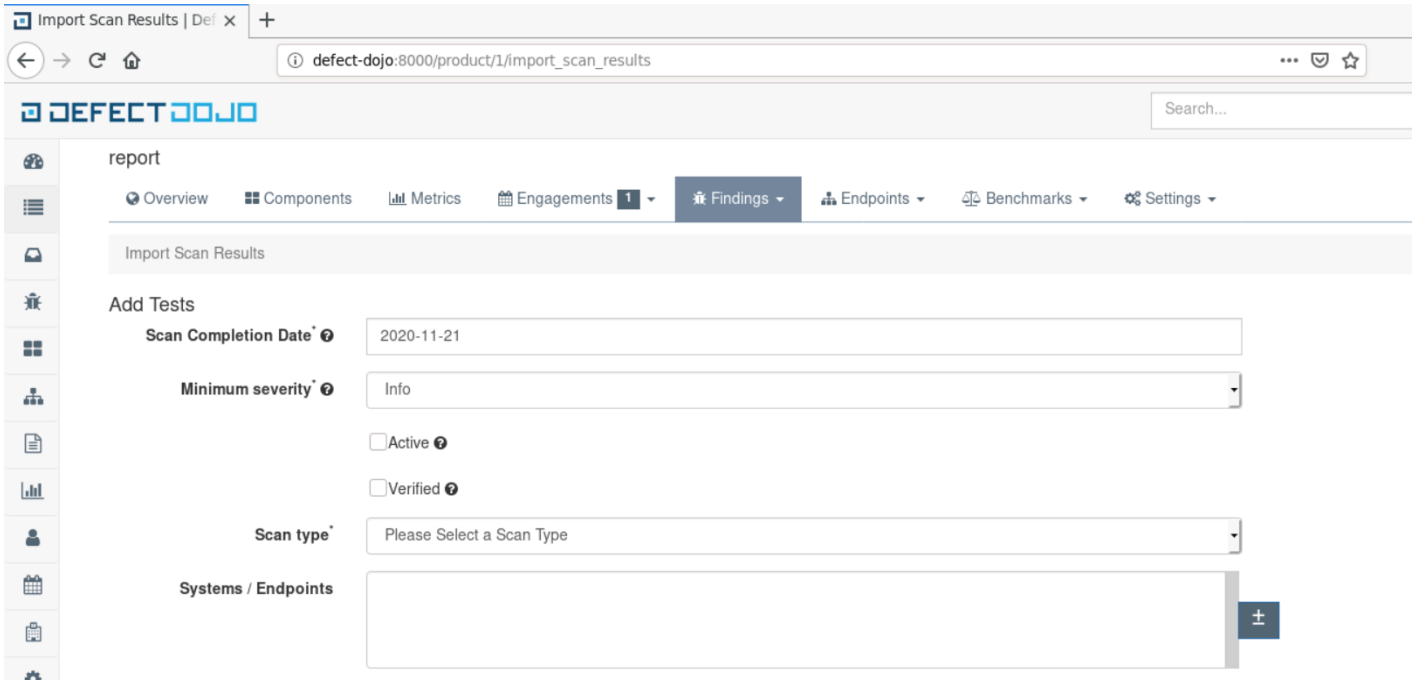


**Issues Detected**

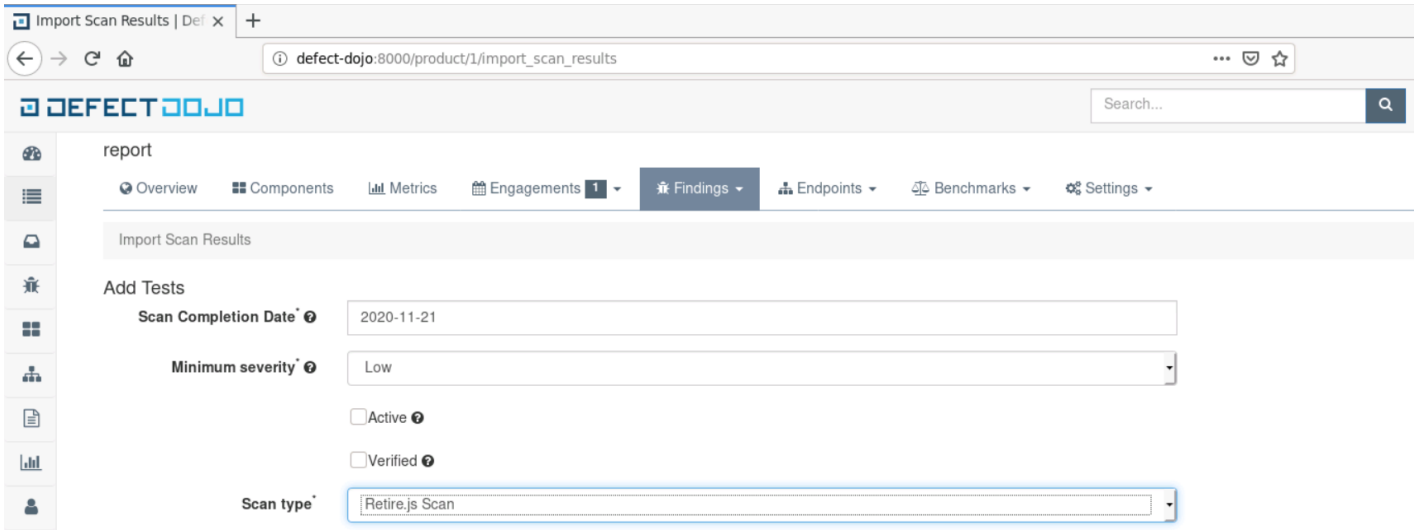
- dojo 1.3.3 is vulnerable to CVE-2010-2275, CVE-2018-15494, CVE-2020-2275, CVE-2020-5258

**Example 2: node-app-template**

**Step 1:** Navigate back to the 'Import Scan Results' page




**Step 2:** Fill all the information such as Minimum severity and scan type




**Step 3:** Open the terminal and navigate to the 'node-app-template' project directory

**Commands:**
cd ~/github-repos/node-app-template
ls

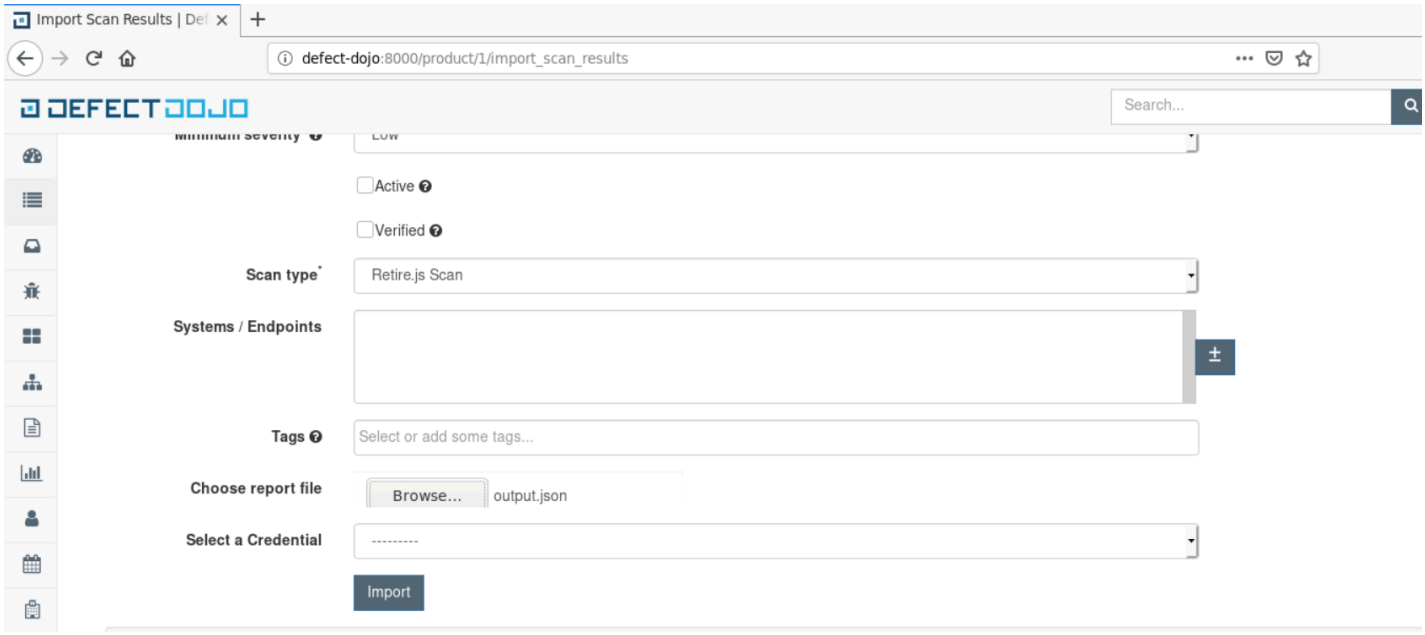
```
root@kali-gui:~/github-repos/is-online# cd ~/github-repos/node-app-template
root@kali-gui:~/github-repos/node-app-template#
root@kali-gui:~/github-repos/node-app-template# ls
LICENSE.md  README.pa  bin         node_modules       package.json  routes
README.md   app.js     config.js   package-lock.json  public        views
root@kali-gui:~/github-repos/node-app-template#
```

**Step 4:** Run the Retire.js scan on the directory and create a JSON file

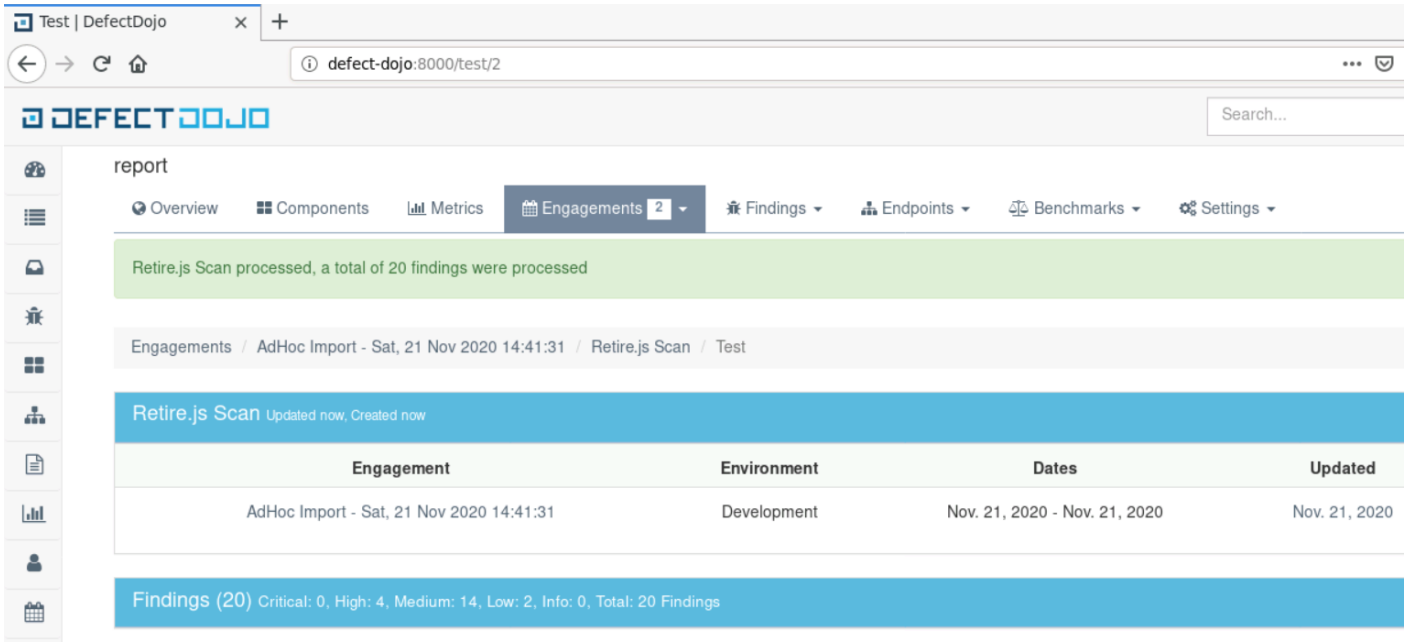
**Commands:**
retire --js --outputformat json  --outputpath output.json
ls

```
root@kali-gui:~/github-repos/node-app-template# retire --js --outputformat json  --outputpath output.json
root@kali-gui:~/github-repos/node-app-template# ls
LICENSE.md  README.pa  bin         node_modules  package-lock.json  public  views
README.md   app.js     config.js   output.json   package.json       routes
root@kali-gui:~/github-repos/node-app-template#
```

**Step 5:** Navigate back to the firefox and upload the output.json file generated via retire.js tool.

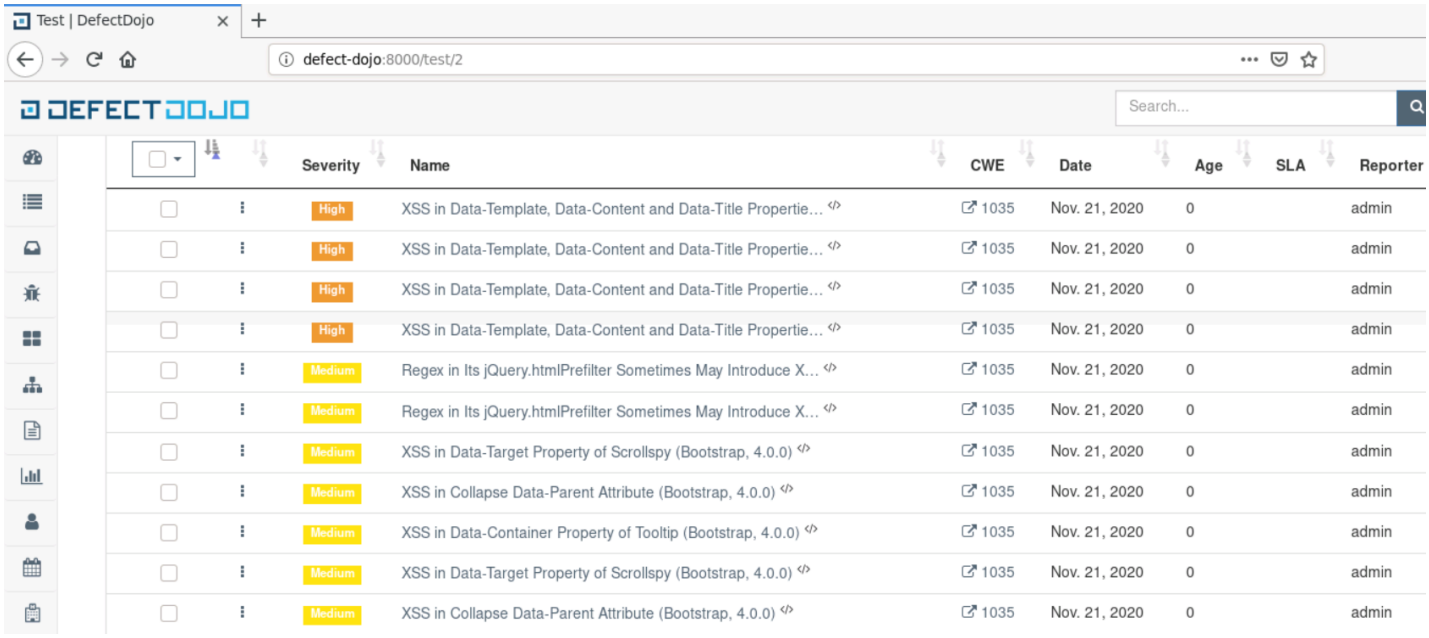



**Step 6:** Click on the Import button to upload the scan results.



**Note:** The findings by the tool may vary in number since there are possible false positives in the results.

Scroll down to check the vulnerabilities found.




**Issues Detected**

- jquery 3.3.1 is vulnerable to CVE-2020-11022 and CVE-2020-11023
- bootstrap 4.0.0 is vulnerable to CVE-2019-8331, CVE-2018-14040, CVE-2018-14041, and CVE-2018-14042

## Learning

- Learning the usage of defect dojo using output generated by retire js

**References:**

- Is-online (https://github.com/sindresorhus/is-online.git)
- Node-app-template (https://github.com/lresende/node-app-template.git)