

ATTACK DEFENSE

by PentesterAcademy

ATTACK DEFENSE LABS COURSES
PENTESTER ACADEMY TOOL BOX PENTESTING
JOINT WORLD-CLASS TRAINERS TRAINING HACKER
TOOL BOX PATV HACKER
HACKER PENTESTING
PATV RED TEAM LABS ATTACK DEFENSE LABS
TRAINING COURSES ACCESS POINT PENTESTER
TEAM LABS PENTESTER ACADEMY ATTACK DEFENSE LABS
GACCESS POINT TOOL BOX WORLD-CLASS TRAINERS
WORLD-CLASS TRAINERS
ATTACK DEFENSE LABS TRAINING COURSES PATV ACCESS
PENTESTER ACADEMY TOOL BOX PENTESTING
ATTACK DEFENSE LABS TRAINING COURSES PENTESTER ACADEMY
COURSES PENTESTER ACADEMY TOOL BOX PENTESTING
TOOL BOX HACKER PENTESTING
PATV RED TEAM LABS ATTACK DEFENSE LABS
COURSES PENTESTER ACADEMY
PENTESTER ACADEMY ATTACK DEFENSE LABS
WORLD-CLASS TRAINERS
RED TEAM TRAINING COURSES
PENTESTER ACADEMY TOOL BOX
PENTESTING

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.

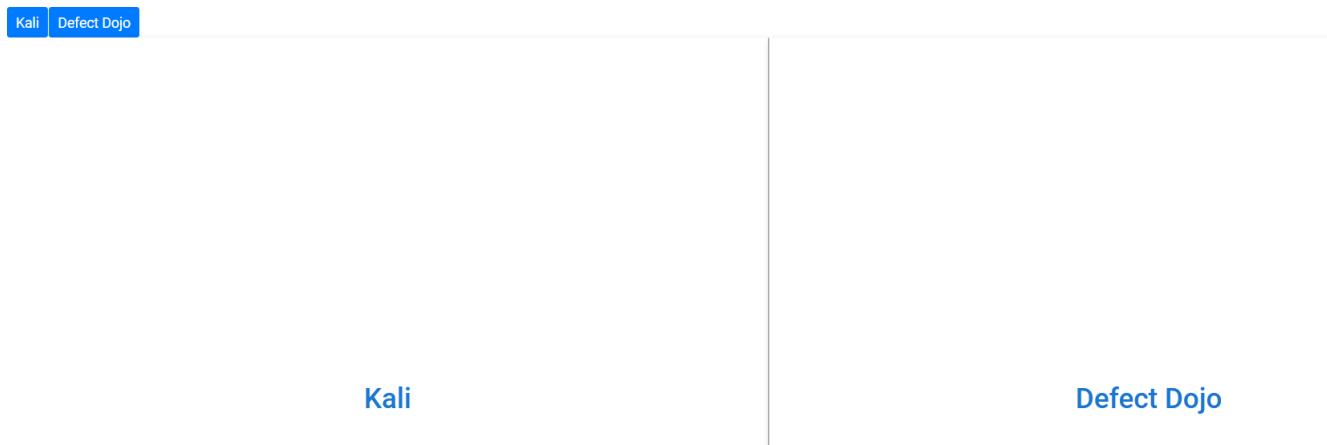
Objective: Follow the manual and learn how to use the Defect-dojo tool!

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

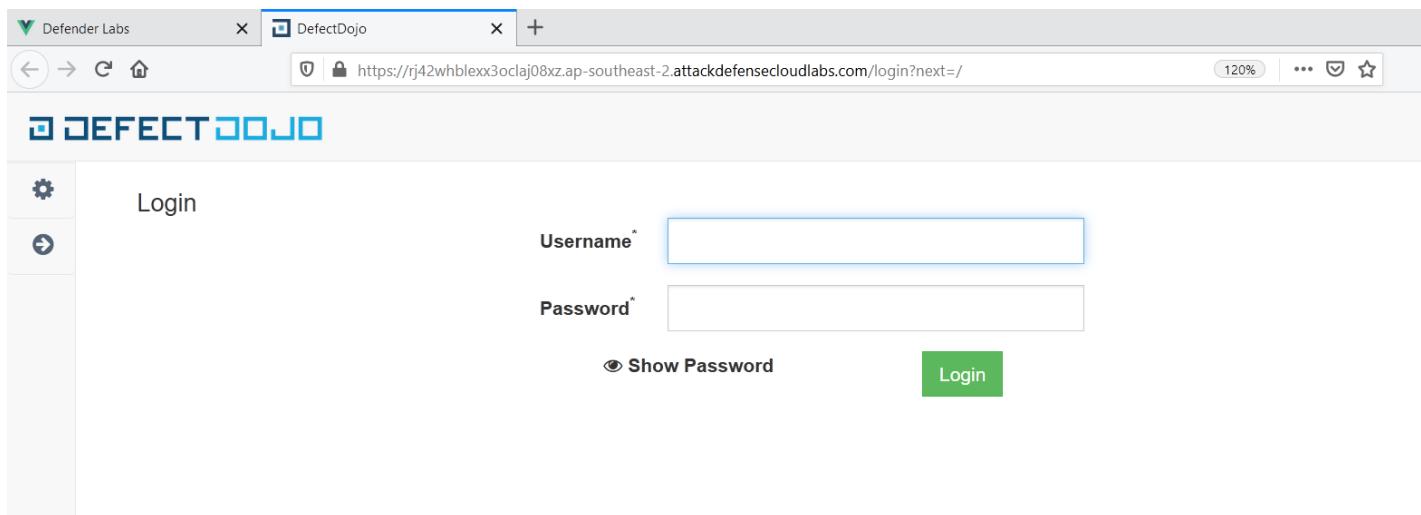
On starting the lab, the following interface will be accessible to the user.



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.

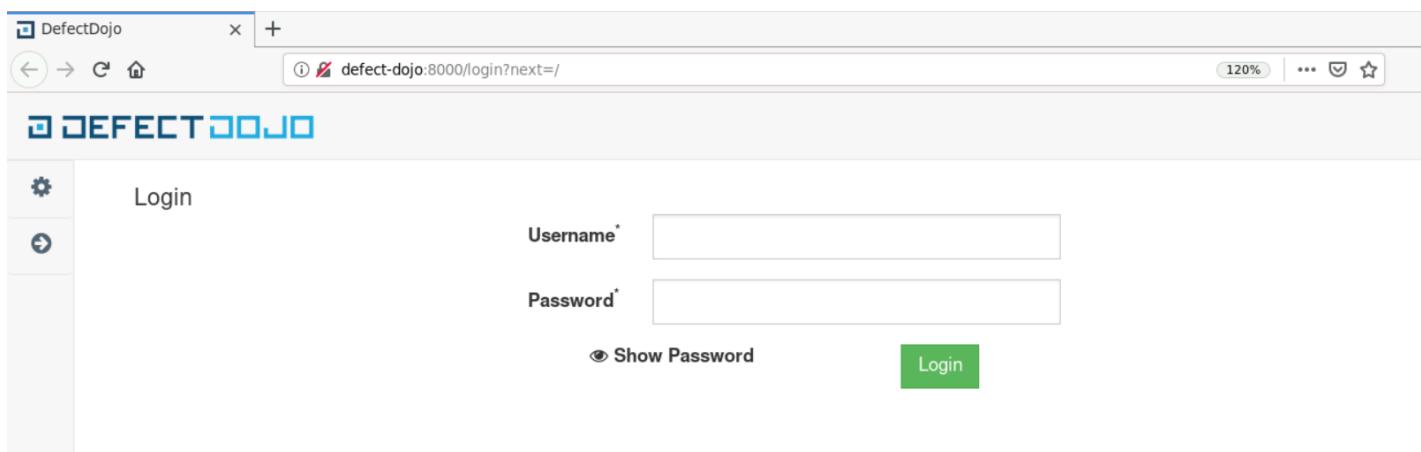


Now, both interfaces are ready and the user can start with the lab.

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

The screenshot shows the DefectDojo dashboard with the following statistics:

- Active Engagements: 0
- Last Seven Days: 0
- Closed In Last Seven Days: 0

Below these statistics are two charts:

- Historical Finding Severity: A chart showing the distribution of findings over time, with a legend indicating Critical (red), Major (orange), Moderate (yellow), and Low (green).
- Reported Finding Severity by Month: A chart showing the distribution of findings by month, with a legend indicating Critical (red), Major (orange), Moderate (yellow), and Low (green). The chart shows a value of 1.0.

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

The screenshot shows the DefectDojo dashboard with a modal overlay displaying product management options:

- Add Product
- All Products
- View Product Types

Below the modal are the same statistics and charts as the previous screenshot.

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

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

New Product | DefectDojo x +

← → C ⌂ 120%

defect-dojo:8000/product/add

DEFECTDOJO

Search...

Home / New Product

Name* report

Description* report

B I H | “ “ = = | ☰ ☱ | ☰ - </> | ?

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

New Product | DefectDojo

defect-dojo:8000/product/add

DEFECT DOJO

Search...

Technical contact

Team manager

Product Type *

Research and Development

Regulations

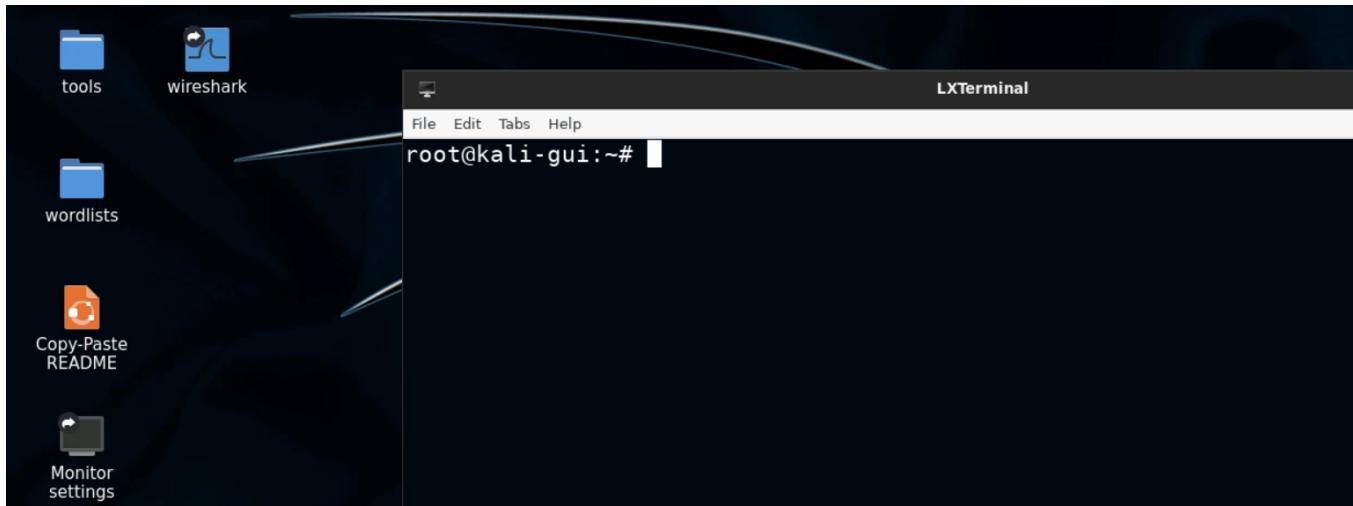
OPPA (United States, California)
CA SB-1386 (United States, California)
COPPA (United States)
DPA (United Kingdom)
Directive 95/46/EC (European Union)

Technologies

Scroll down and click on the “Submit” button.

The screenshot shows the DefectDojo interface. The top navigation bar includes a back button, forward button, search bar ('defect-dojo:8000/product/1'), zoom level (120%), and a star icon. The main header is 'DEFECTDOJO' with a search bar. On the left, there's a sidebar with various icons. The main content area has a title 'report'. Below it is a success message: 'Product added successfully.' A 'Description' section contains the word 'report'. At the bottom, a 'Metrics' section displays counts for different severity levels: CRITICAL (0), HIGH (0), MEDIUM (0), LOW (0), INFORMATIONAL (0), and TOTAL (0).

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.md  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#

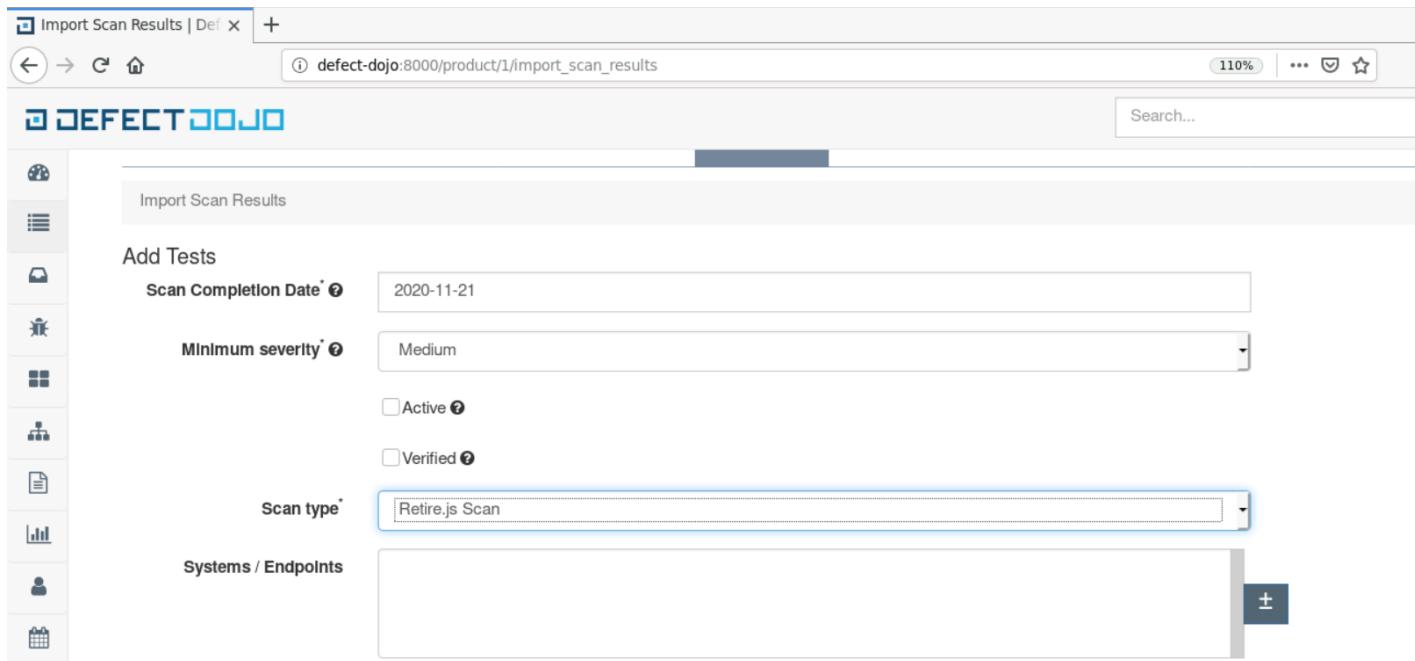
```

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

The screenshot shows the DefectDojo application interface. The top navigation bar includes tabs for Product, DefectDojo, and a plus sign icon. Below the navigation is a toolbar with icons for back, forward, search, and zoom. The main content area has a title 'DEFECTDOJO' and a sidebar with icons for report, components, metrics, engagements, findings, endpoints, benchmarks, and settings. The 'Findings' tab is currently active. The main panel displays a 'Description' section with the word 'report' and a 'Metrics' section showing counts for Critical (0), High (0), Medium (0), and Low (0) severity levels. To the right, a sidebar provides links to various findings-related functions and metadata fields like Business Criticality, Product Type, Platform, Lifecycle, and Origin.

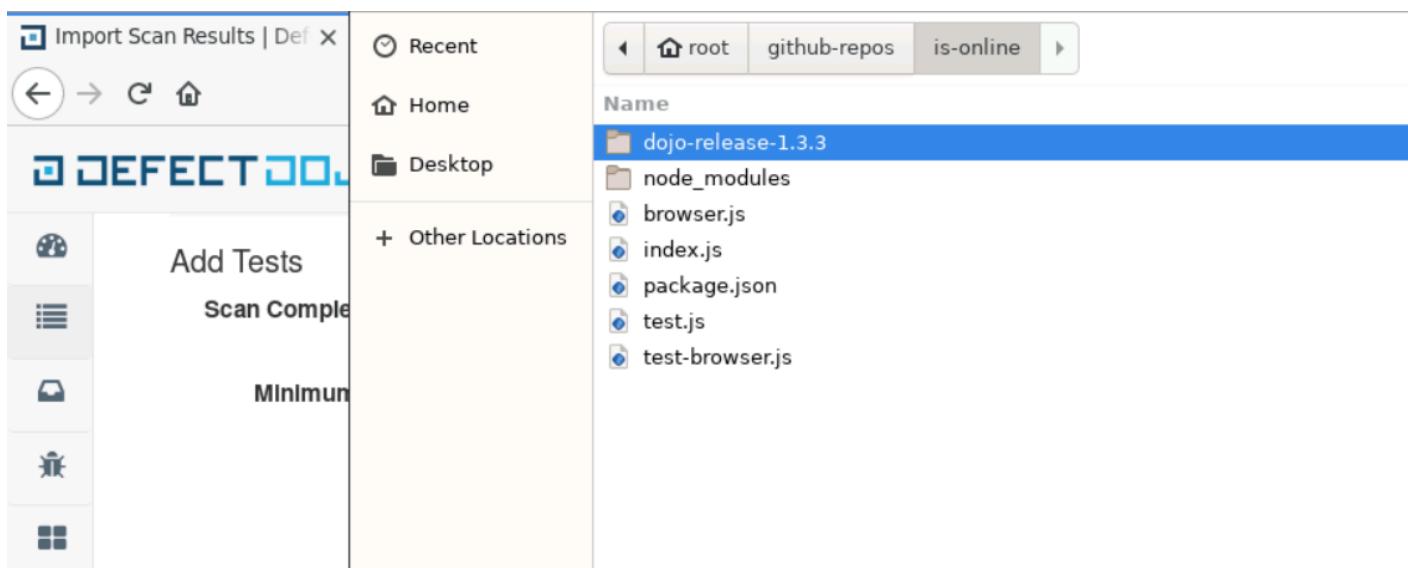
This screenshot shows the 'Import Scan Results' page within the DefectDojo application. The top navigation bar is visible with the 'Import Scan Results' tab selected. The main form contains fields for 'Scan Completion Date' (set to 2020-11-21), 'Minimum severity' (set to Info), and checkboxes for 'Active' and 'Verified'. There is also a dropdown menu for 'Scan type' with the placeholder 'Please Select a Scan Type'. On the left, there is a sidebar with icons for report, components, metrics, engagements, findings, endpoints, benchmarks, and settings.

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



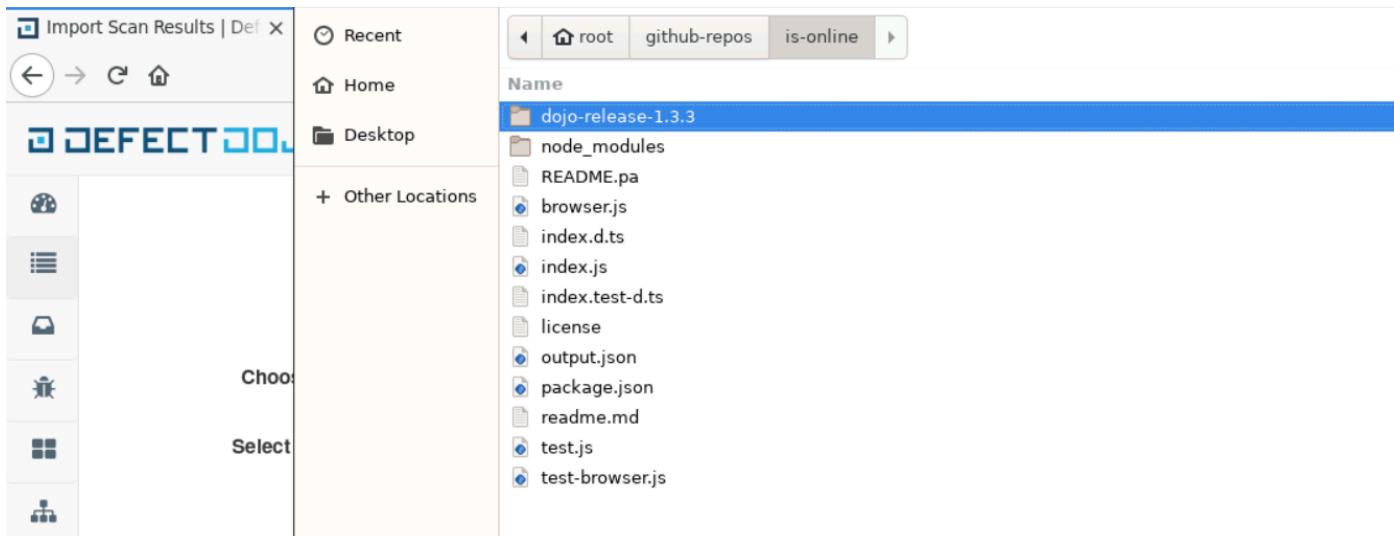
The screenshot shows the 'Import Scan Results' page of the DEFECT DOJO application. On the left, there is a sidebar with various icons. The main area has a title 'Import Scan Results'. Below it, there's a form for 'Add Tests'. The 'Scan Completion Date' field contains '2020-11-21'. The 'Minimum severity' dropdown is set to 'Medium'. There are two unchecked checkboxes: 'Active' and 'Verified'. The 'Scan type' dropdown is set to 'Retire.js Scan'. Below the form is a section titled 'Systems / Endpoints' with a large input field and a plus sign icon.

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

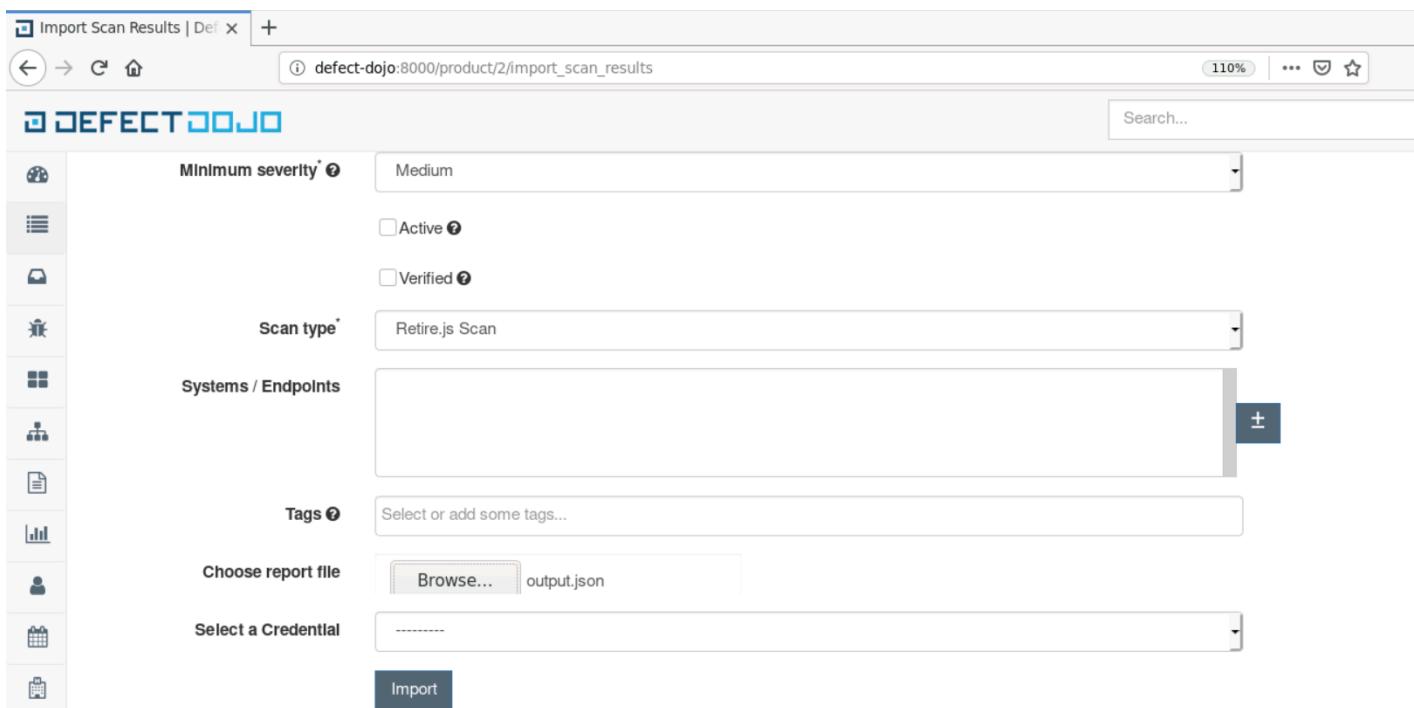


The screenshot shows the 'Import Scan Results' page with a file upload dialog overlaid on the sidebar. The sidebar icons are visible on the left. The main area shows the 'Add Tests' form with the 'Scan type' set to 'Retire.js Scan'. The file upload dialog shows a list of files from a directory named 'dojo-release-1.3.3'. The files listed are 'node_modules', 'browser.js', 'index.js', 'package.json', 'test.js', and 'test-browser.js'. The 'index.js' file is highlighted with a blue selection bar.

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.

The screenshot shows the DefectDojo web application. The URL in the browser is `defect-dojo:8000/test/1`. The main navigation bar includes links for Overview, Components, Metrics, Engagements (selected), Findings, Endpoints, Benchmarks, and Settings. A search bar is located at the top right. On the left, there is a sidebar with various icons. The main content area displays a message: "Retire.js Scan processed, a total of 8 findings were processed". Below this, the breadcrumb navigation shows: Engagements / AdHoc Import - Sat, 21 Nov 2020 14:31:30 / Retire.js Scan / Test. A table titled "Retire.js Scan" shows one engagement entry: "AdHoc Import - Sat, 21 Nov 2020 14:31:30" with environment "Development", dates "Nov. 21, 2020 - Nov. 21, 2020", and updated on "Nov. 21, 2020". At the bottom, it says "Findings (8) Critical: 0, High: 0, Medium: 8, Low: 0, Info: 0, Total: 8 Findings".

Scroll down to check the vulnerabilities

The screenshot shows the DefectDojo web application. The URL in the browser is `defect-dojo:8000/test/1`. The main navigation bar includes links for Overview, Components, Metrics, Engagements, Findings (selected), Endpoints, Benchmarks, and Settings. A search bar is located at the top right. On the left, there is a sidebar with various icons. The main content area displays a table titled "Findings (8) Critical: 0, High: 0, Medium: 8, Low: 0, Info: 0, Total: 8 Findings". The table has columns for Column visibility, Copy, Excel, CSV, PDF, Print, Severity, Name, CWE, Date, Age, SLA, Reporter, and Status. The data in the table is as follows:

Severity	Name	CWE	Date	Age	SLA	Reporter	Status
Medium	CVE-2010-2275 (Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	(Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	CVE-2018-15494 (Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	CVE-2020-5258 (Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	CVE-2010-2275 (Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	(Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	CVE-2018-15494 (Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive
Medium	CVE-2020-5258 (Dojo, 1.3.3) ↗	1035	Nov. 21, 2020	0		admin	Inactive

Issues Detected

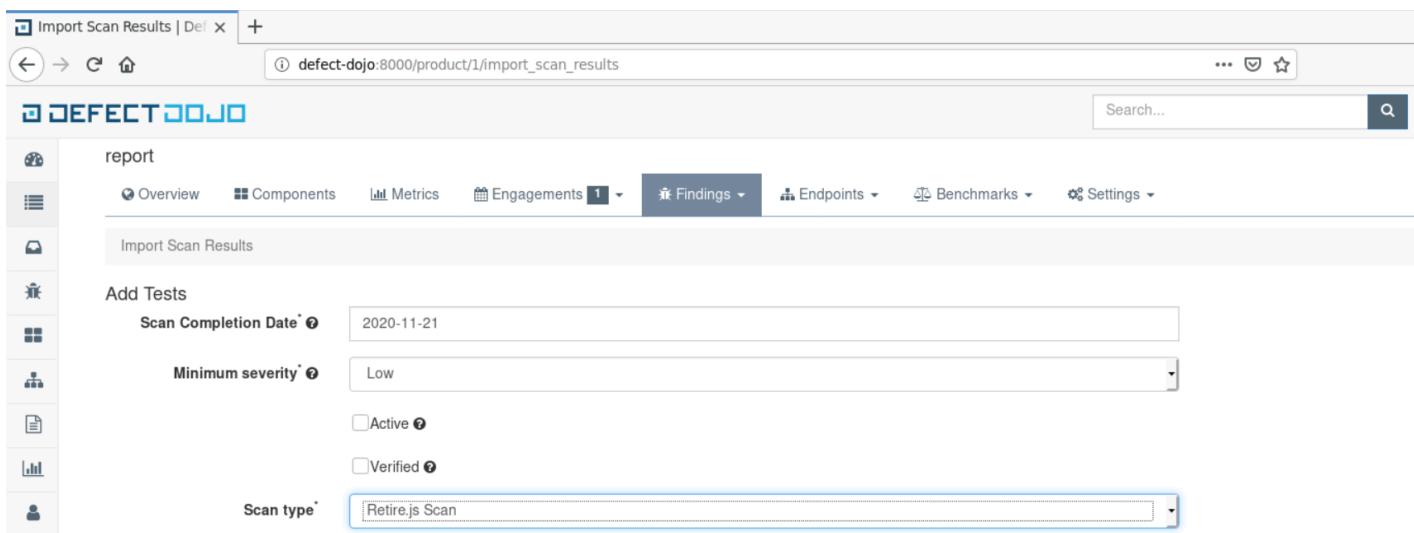
- dojo 1.3.3 is vulnerable to CVE-2010-2275, CVE-2018-15494, CVE-2020-2275, CVE-2020-5258

Example 2: node-app-template

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

The screenshot shows the DEFECTDOJO web application interface. The top navigation bar displays the URL `defect-dojo:8000/product/1/import_scan_results`. The main menu includes options like Overview, Components, Metrics, Engagements, Findings (selected), Endpoints, Benchmarks, and Settings. On the left, a sidebar provides quick access to various reports. The central form is titled 'Import Scan Results' and contains fields for 'Scan Completion Date' (set to 2020-11-21), 'Minimum severity' (set to Info), and 'Scan type' (a dropdown menu showing 'Please Select a Scan Type').

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.

Import Scan Results | Def x +

defect-dojo:8000/product/1/import_scan_results

DEFECTDOJO

Search...

Minimum Severity: Low

Active

Verified

Scan type*: Retire.js Scan

Systems / Endpoints

Tags ? Select or add some tags...

Choose report file output.json

Select a Credential

Import

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

Test | DefectDojo x +

defect-dojo:8000/test/2

DEFECTDOJO

report

Overview Components Metrics Engagements 2 Findings Endpoints Benchmarks Settings

Retire.js Scan processed, a total of 20 findings were processed

Engagements / AdHoc Import - Sat, 21 Nov 2020 14:41:31 / Retire.js Scan / Test

Engagement	Environment	Dates	Updated
AdHoc Import - Sat, 21 Nov 2020 14:41:31	Development	Nov. 21, 2020 - Nov. 21, 2020	Nov. 21, 2020

Findings (20) Critical: 0, High: 4, Medium: 14, Low: 2, Info: 0, Total: 20 Findings

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.

	<input type="checkbox"/>	Severity	Name	CWE	Date	Age	SLA	Reporter
	<input type="checkbox"/>	High	XSS in Data-Template, Data-Content and Data-Title Properties (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	High	XSS in Data-Template, Data-Content and Data-Title Properties (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	High	XSS in Data-Template, Data-Content and Data-Title Properties (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	High	XSS in Data-Template, Data-Content and Data-Title Properties (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	Regex in Its jQuery.htmlPrefilter Sometimes May Introduce XSS (jQuery, 3.3.1)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	Regex in Its jQuery.htmlPrefilter Sometimes May Introduce XSS (jQuery, 3.3.1)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	XSS in Data-Target Property of Scrollspy (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	XSS in Collapse Data-Parent Attribute (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	XSS in Data-Container Property of Tooltip (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	XSS in Data-Target Property of Scrollspy (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin
	<input type="checkbox"/>	Medium	XSS in Collapse Data-Parent Attribute (Bootstrap, 4.0.0)	1035	Nov. 21, 2020	0	admin	admin

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/iresende/node-app-template.git>)