



SECURITY+ V4 LAB SERIES

Lab 6: Vulnerability Checks with OpenVAS

Document Version: **2022-04-29**

Material in this Lab Aligns to the Following	
CompTIA Security+ (SY0-601) Exam Objectives	1.6: Explain the security concerns associated with various types of vulnerabilities 1.7: Summarize the techniques used in security assessments
All-In-One CompTIA Security+ Sixth Edition ISBN-13: 978-1260464009 Chapters	6: Vulnerabilities 7: Security Assessments

Copyright © 2022 Network Development Group, Inc.
www.netdevgroup.com

NETLAB+ is a registered trademark of Network Development Group, Inc.
KALI LINUX™ is a trademark of Offensive Security.
Microsoft®, Windows®, and Windows Server® are trademarks of the Microsoft group of companies.
VMware is a registered trademark of VMware, Inc.
SECURITY ONION is a trademark of Security Onion Solutions LLC.
Android is a trademark of Google LLC.
pfSense® is a registered mark owned by Electric Sheep Fencing LLC ("ESF").
All trademarks are property of their respective owners.

Contents

Introduction	3
Objective	3
Lab Topology	4
Lab Settings	5
1 Conduct a Vulnerability Check Using OpenVAS.....	6
1.1 Start and Set the Services	6
1.2 Start the OpenVAS Scanner.....	10
2 Scan the Vulnerability Using OpenVAS.....	12
2.1 Set the Target	12
2.2 Add a Task	13
2.3 Scan Vulnerabilities and Check Reports.....	14

Introduction

In this lab, you will use open-source tools that are available on the internet to check vulnerabilities.

Objective

In this lab, you will perform the following tasks:

- Conduct vulnerability check using OpenVAS

Lab Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account (if needed)	Password (if needed)
Kali	203.0.113.2	kali	kali

1 Conduct a Vulnerability Check Using OpenVAS

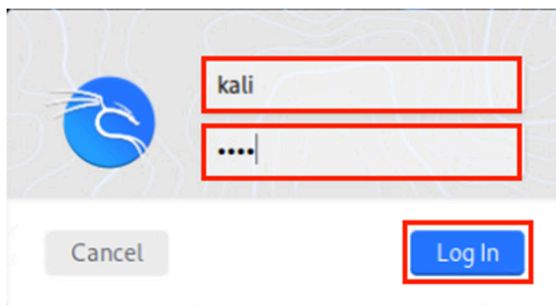
1.1 Start and Set the Services

In this section, you will start the services required to perform the lab activities.

1. Launch the **Kali** virtual machine to access the graphical login screen.



2. Log in as **kali** with the password **kali**.

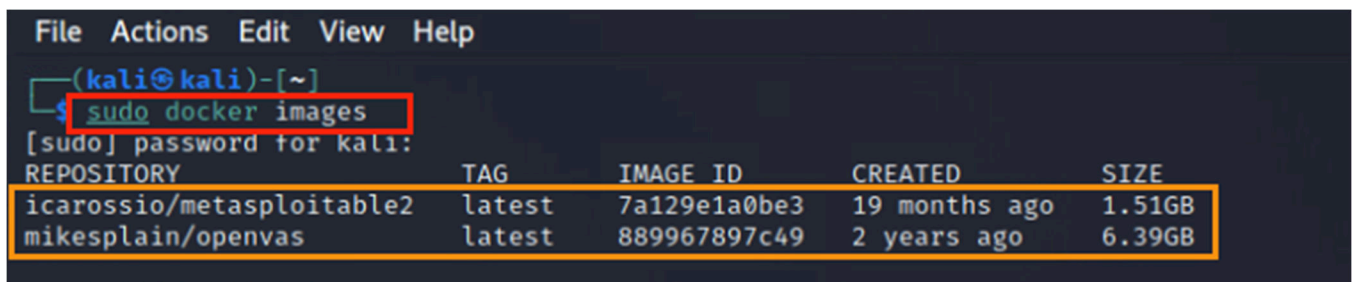


3. Open a *Terminal* window by clicking on the **Terminal** icon, located in the left menu pane.



4. Enter the command below to check the available docker images. When prompted for the password, type **kali**. Verify that **icarossio/metasploitable2** and **mikesplain/openvas** exist.

```
kali@kali$ sudo docker images
```

A screenshot of a terminal window. The title bar shows 'File Actions Edit View Help'. The prompt is '(kali@kali)-[~]'. The command 'sudo docker images' is entered and highlighted with a red box. Below it, the prompt '[sudo] password for kali:' is shown. The output of the command is displayed in a table with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The first two rows are highlighted with an orange box.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
icarossio/metasploitable2	latest	7a129e1a0be3	19 months ago	1.51GB
mikesplain/openvas	latest	889967897c49	2 years ago	6.39GB

- Next, let's start and set the *metasploitable* service. In the *Terminal* window, type the following command to start the *metasploit2*. This command will forward every port to the host kali machine. Once executed, you will see a long string of numbers and letters. This will be the container-id of the next step.

```
kali@kali$ sudo docker run --rm -ditP icarossio/metasploitable2
```

```
(kali@kali)-[~]  
$ sudo docker run --rm -ditP icarossio/metasploitable2  
4c64bd80aac19cfefb325f4cfb0dbd8ec5888e6ac428da1a3dab2dbe0978f47e
```

- We will now use the container-id to check the mapped ports between the kali machine and the container. Type the command below to check all mapped ports. Notice that we only used the first 4 characters/numbers from the container-id. All the mapped ports are within the 49153-49172 range.

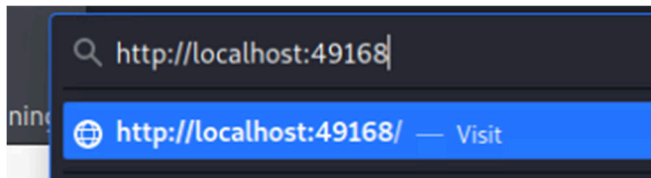
```
kali@kali$ sudo docker port 4c64
```

```
(kali@kali)-[~]  
$ sudo docker port 4c64  
25/tcp → 0.0.0.0:49169  
3632/tcp → 0.0.0.0:49158  
80/tcp → 0.0.0.0:49168  
111/tcp → 0.0.0.0:49167  
2121/tcp → 0.0.0.0:49160  
445/tcp → 0.0.0.0:49165  
6667/tcp → 0.0.0.0:49154  
1524/tcp → 0.0.0.0:49161  
21/tcp → 0.0.0.0:49172  
513/tcp → 0.0.0.0:49163  
5432/tcp → 0.0.0.0:49157  
5900/tcp → 0.0.0.0:49156  
22/tcp → 0.0.0.0:49171  
512/tcp → 0.0.0.0:49164  
3306/tcp → 0.0.0.0:49159  
514/tcp → 0.0.0.0:49162  
6000/tcp → 0.0.0.0:49155  
8009/tcp → 0.0.0.0:49153  
139/tcp → 0.0.0.0:49166  
23/tcp → 0.0.0.0:49170
```

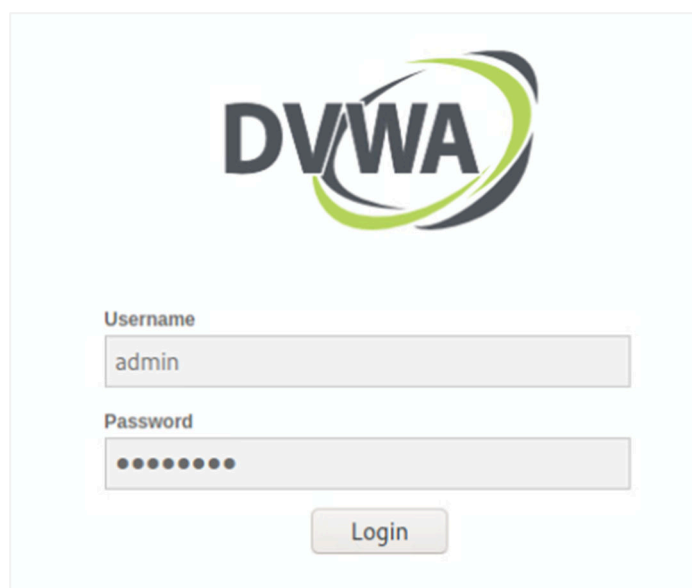
- With the *metasploitable2* running, let's click the **browser icon** to start the browser.



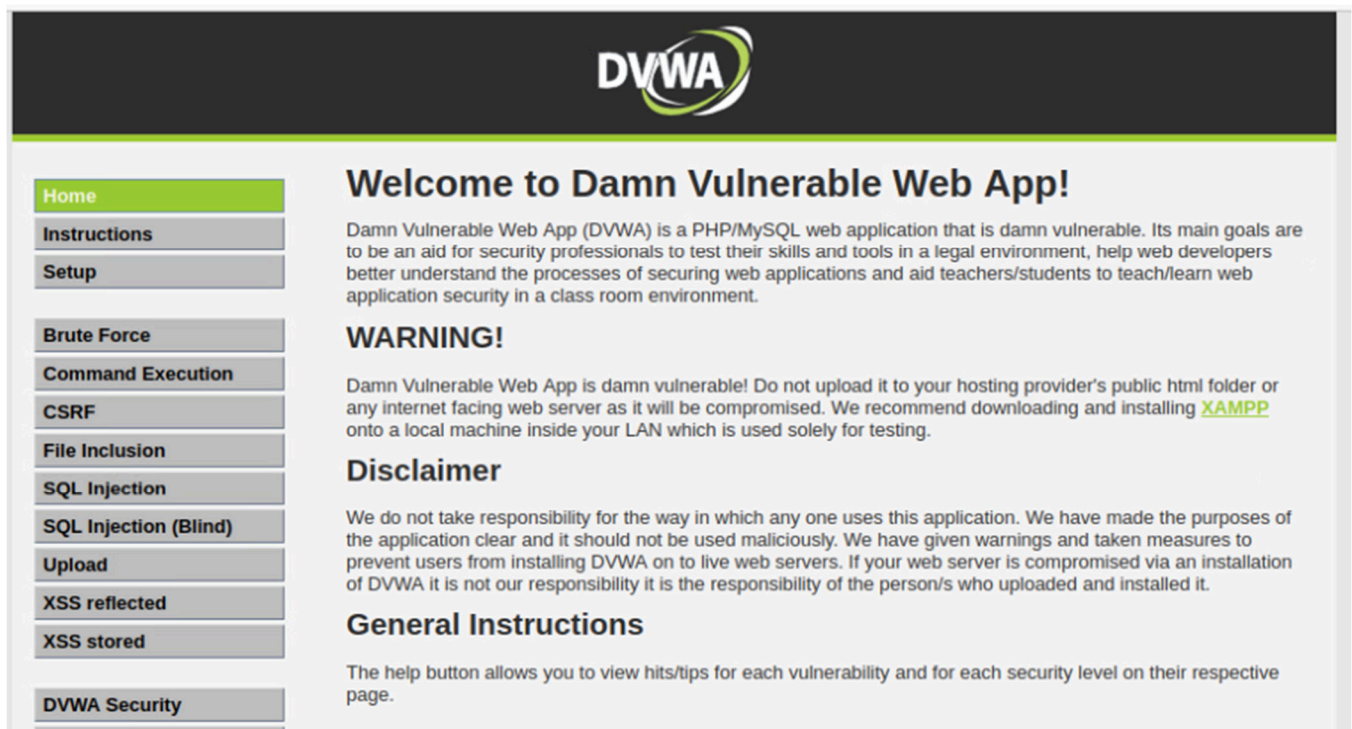
8. In the browser window, click on the address bar, and type the address of `http://localhost:49168`. Press **Enter** and the *metasploitable2* will open.



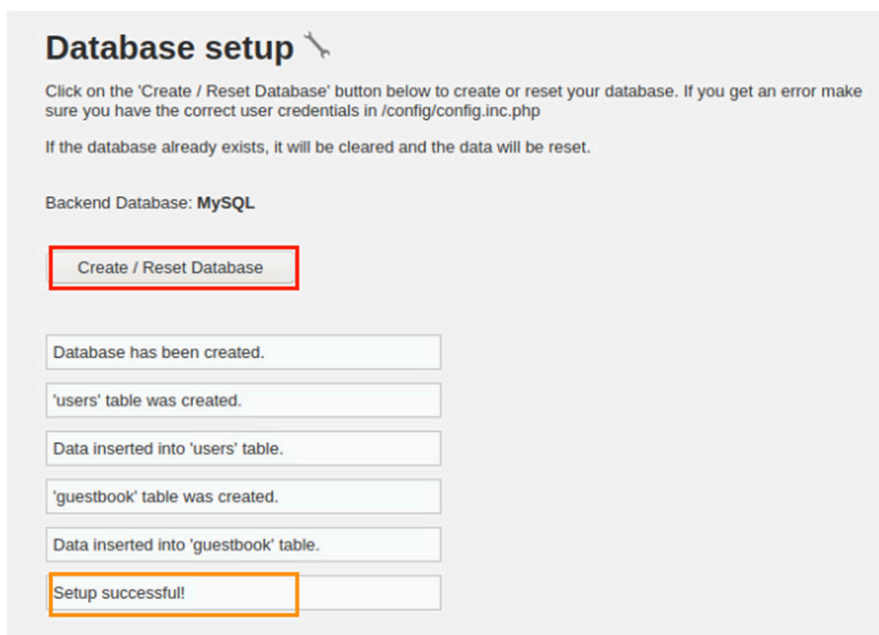
9. On the new page, we will check the database status of the *DVWA* machine. First, click on the **DVWA** link. When the following screen shows up, type **admin** as the username and **password** as the password. Then, click the **Login** button to log in. When prompted to save the password, click **Never**.



10. If you see the following welcome screen, it means the service is up and running.



11. If, for some reason, the service failed to initiate, click the **Setup** button. Then, click **Create/Reset Database**. When finished, you will see the page prompt saying, *Setup successful!*



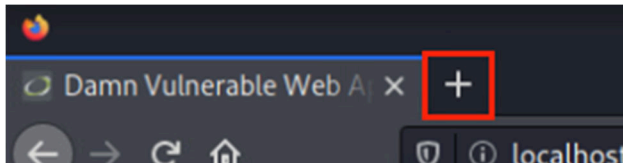
1.2 Start the OpenVAS Scanner

1. Switch back to the *Terminal* window.
2. Type the following command; if prompted for a password, type `kal`.

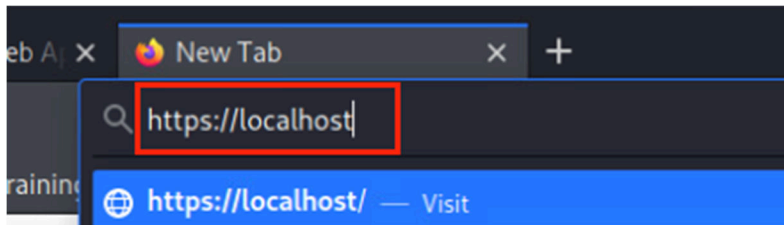
```
kali@kali$ sudo docker run --rm -d -p 443:443 --name openvas mikesplain/openvas
```

```
(kali㉿kali)-[~]
└─$ sudo docker run --rm -d -p 443:443 --name openvas mikesplain/openvas
[sudo] password for kali:
52f5e727ac194e1d31122ed12c9325698b1af064449dceefe6529ae564400cef
```

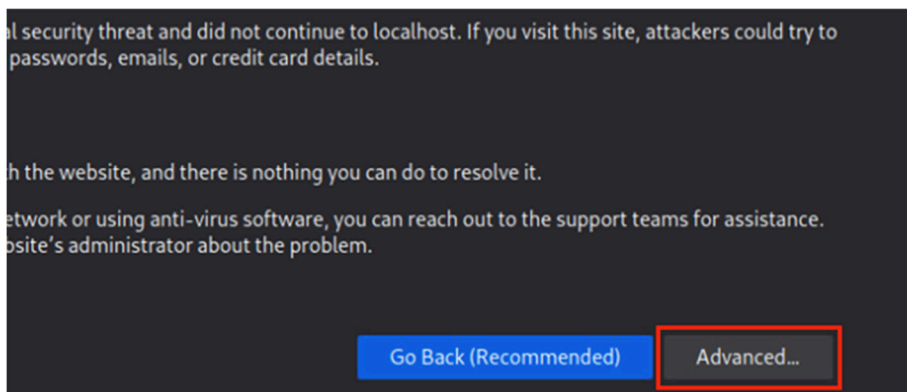
3. Switch back to the browser window; we will check the *OpenVAS* service status.
4. Click the + sign to start a new tab in the browser.



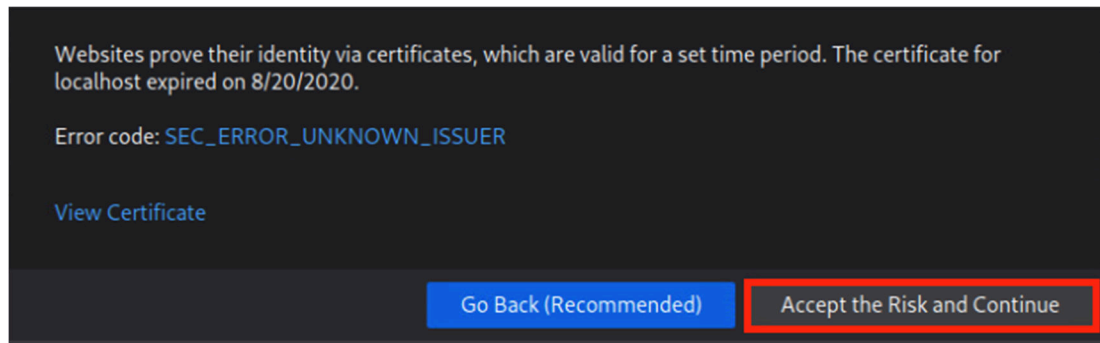
5. In the address bar, type `https://localhost`.



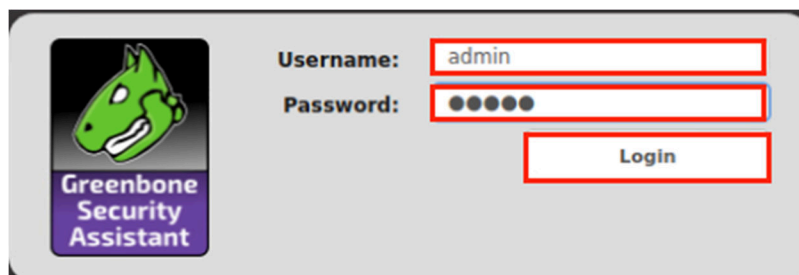
6. On the opened page, click the **Advanced...** button.



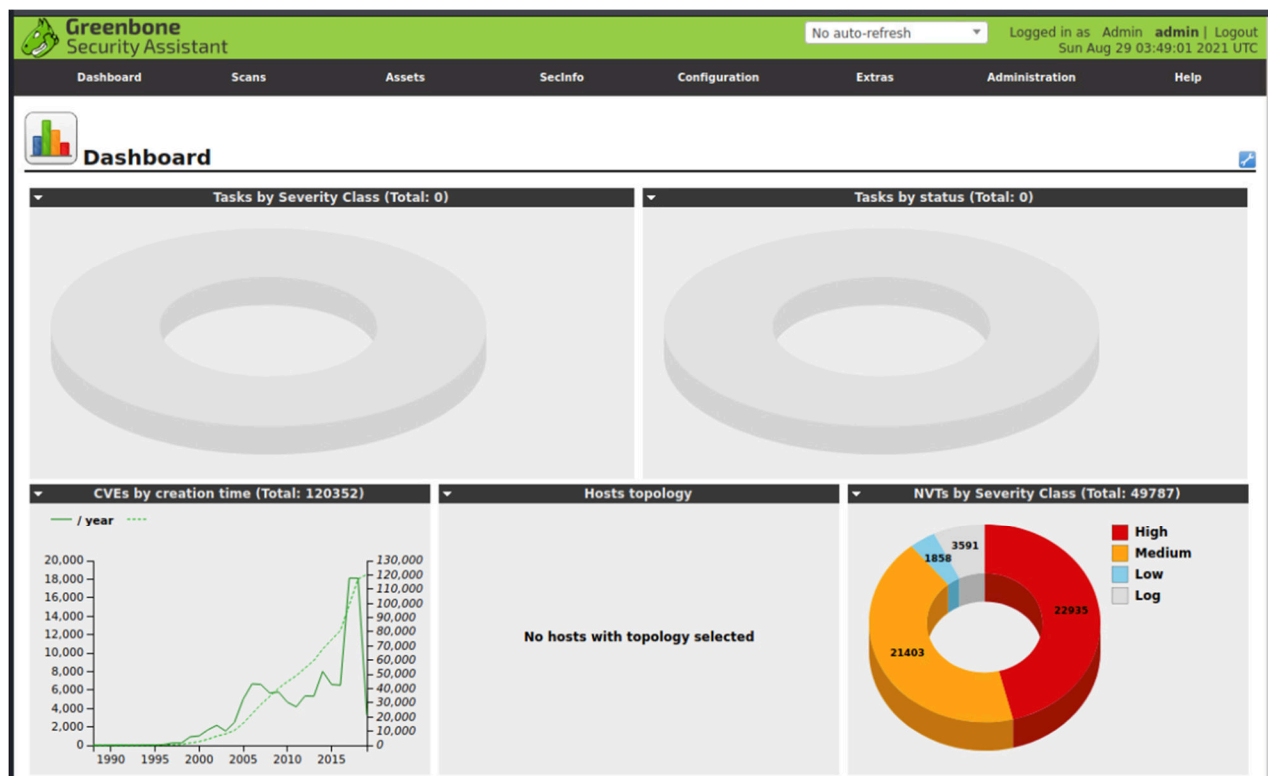
7. A new area will appear; click **Accept the Risk and Continue**.



8. A login page will appear once you click the button. Type **admin** as the username, **admin** as the password. Then, click the **Login** button. When prompted to save the password, click **Don't Save**.



9. You will view the *OpenVAS (Greenbone Security Assistant) Dashboard*.

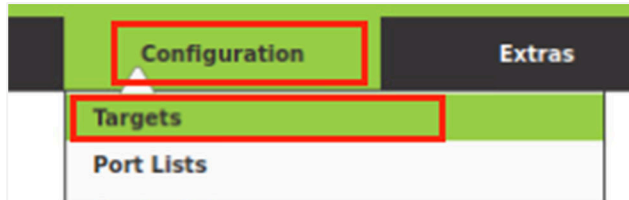


10. Leave the window open, and continue to the next section.

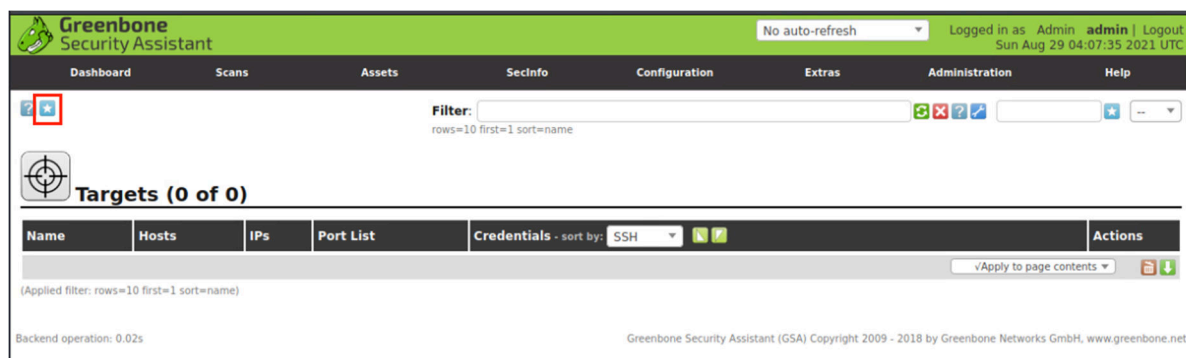
2 Scan the Vulnerability Using OpenVAS

2.1 Set the Target

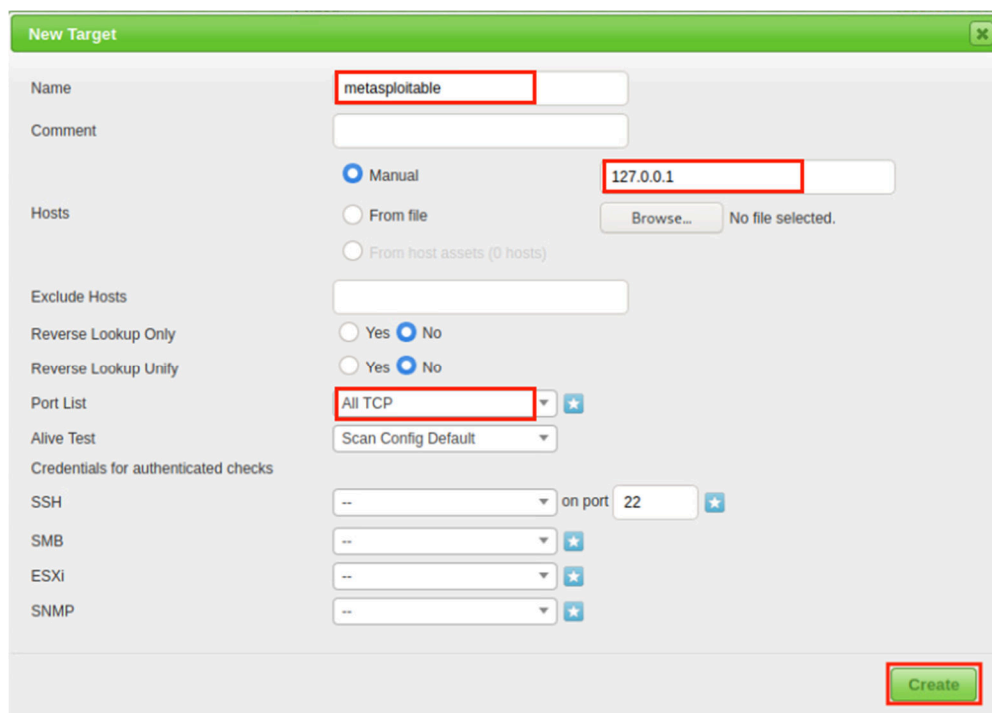
1. In the *OpenVAS* dashboard window, click the **Configuration**, then **Targets**.



2. On the *Targets* page, click the upper-left **star button** to add a target.



3. In the *New Target* window. Type `metasploitable` as the *Name*, add IP address `127.0.0.1` as the target address, and change the *Port List* to **All TCP**. Click the **Create** button once finished. Because docker is forwarding all the ports to the localhost, we are setting the localhost as the target.



New Target

Name:

Comment:

Hosts: ☒ Manual No file selected.

☐ From file

☐ From host assets (0 hosts)

Exclude Hosts:

Reverse Lookup Only: ☐ Yes ☒ No

Reverse Lookup Unify: ☐ Yes ☒ No

Port List:

Alive Test:

Credentials for authenticated checks

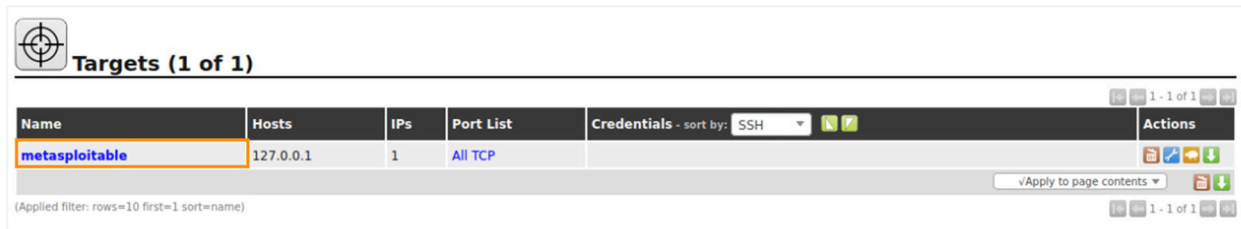
SSH: on port

SMB:

ESXi:

SNMP:

- You will see the target successfully created after clicking the **Create** button in the last step.

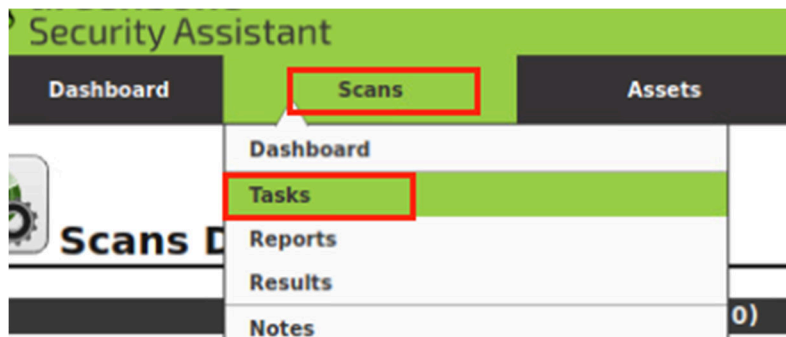


Name	Hosts	IPs	Port List	Credentials - sort by: SSH	Actions
metasploitable	127.0.0.1	1	All TCP		

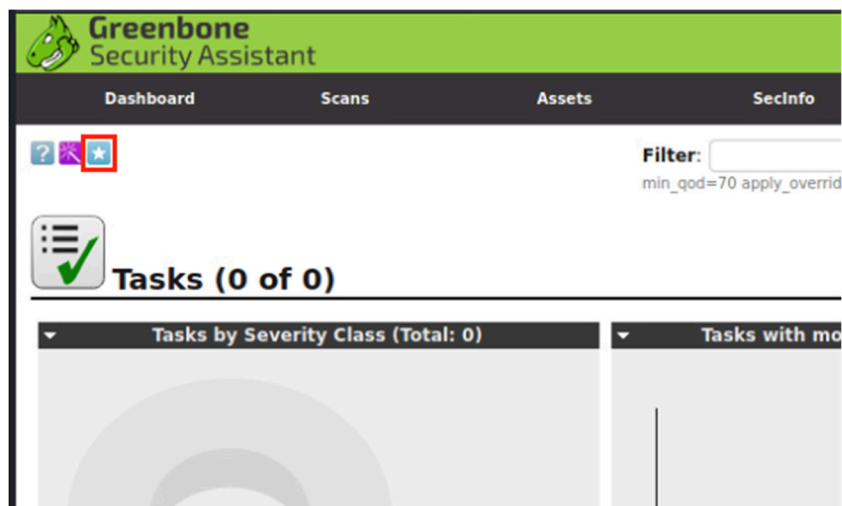
(Applied filter: rows=10 first=1 sort=name)

2.2 Add a Task

- Next, we will create a task to scan the target. Select **Scans > Tasks**.



- Currently, there is no task. Click the **star** button in the upper-left corner to add a new task.



- In the *New Task* window, type the *Name* as **metasploitable scan**, select the **metasploitable** as *Scan Target*, then select **OpenVAS Default** as the *Scanner* and **Full and fast** as the *Scan Config*. Click **Create** when finished.

New Task

Name

metasploitable scan

Comment

Scan Targets

metasploitable

Alerts

Schedule

--

Once

Add results to Assets

yes

no

Apply Overrides

yes

no

Min QoD

70

%

Alterable Task

yes

no

Auto Delete Reports

Do not automatically delete reports

Automatically delete oldest reports but always keep newest

5

reports

Scanner

OpenVAS Default

Scan Config

Full and fast

Network Source Interface

Order for target hosts

Sequential

Maximum concurrently executed NVTs per host

4

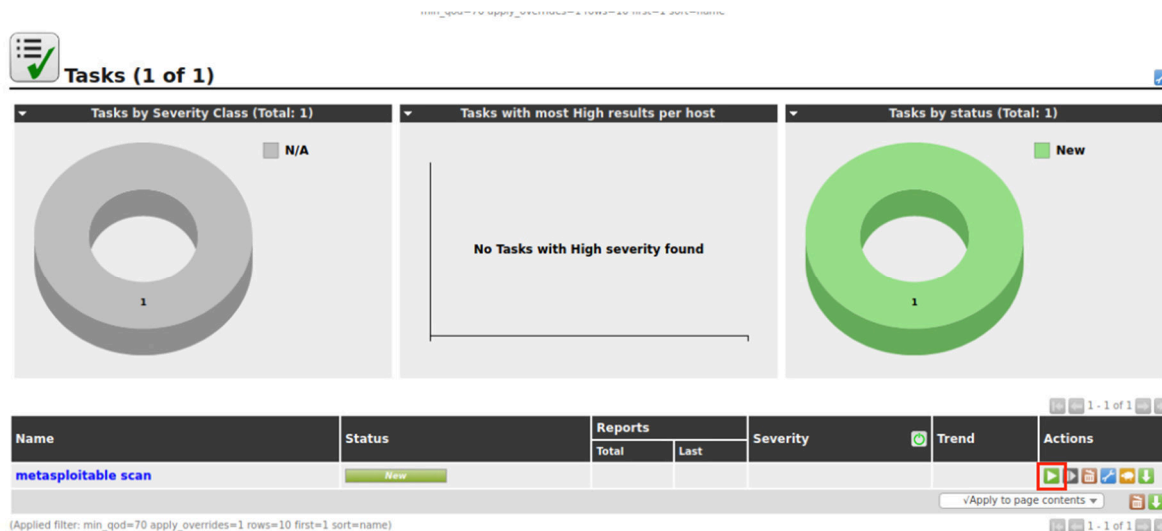
Maximum concurrently scanned hosts

20

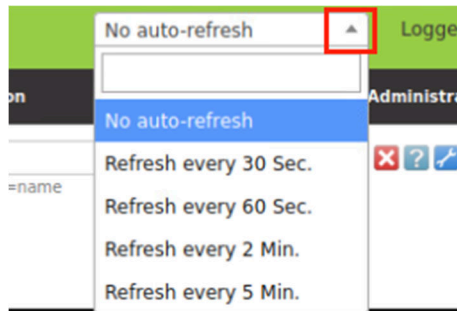
Create

2.3 Scan Vulnerabilities and Check Reports

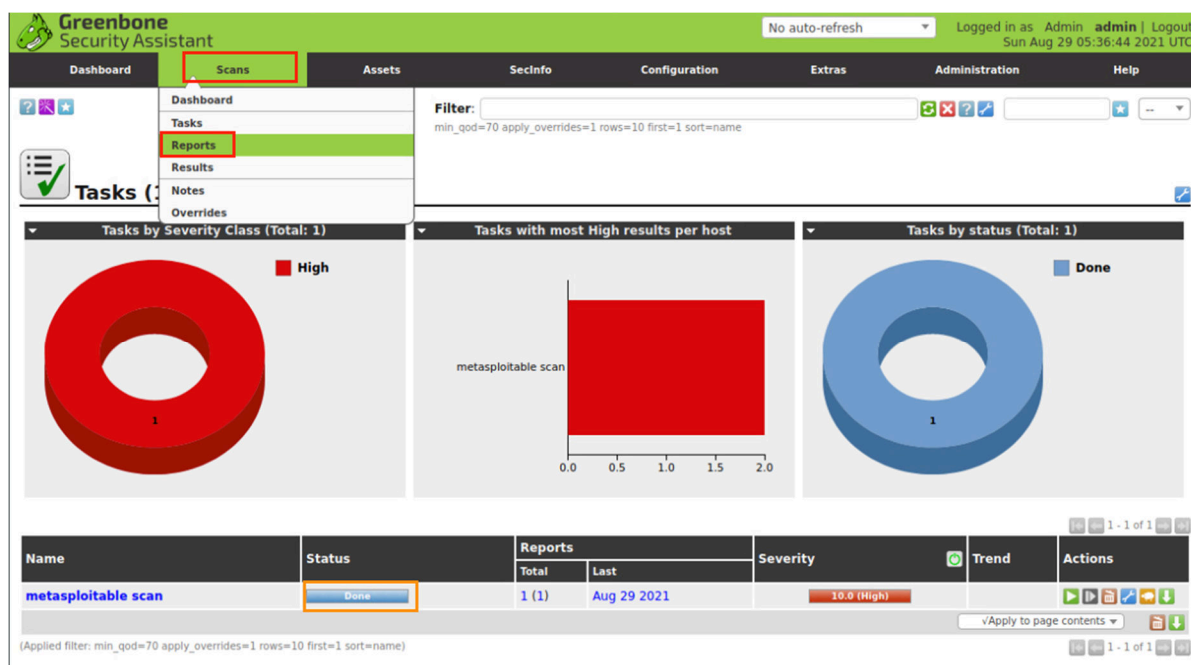
- Now you should see the newly created scan task appear in the *Tasks* window. To the lower right-hand corner, click the **Start** button to start the scan. The scan process may take 10 - 30 minutes. So, please be patient.



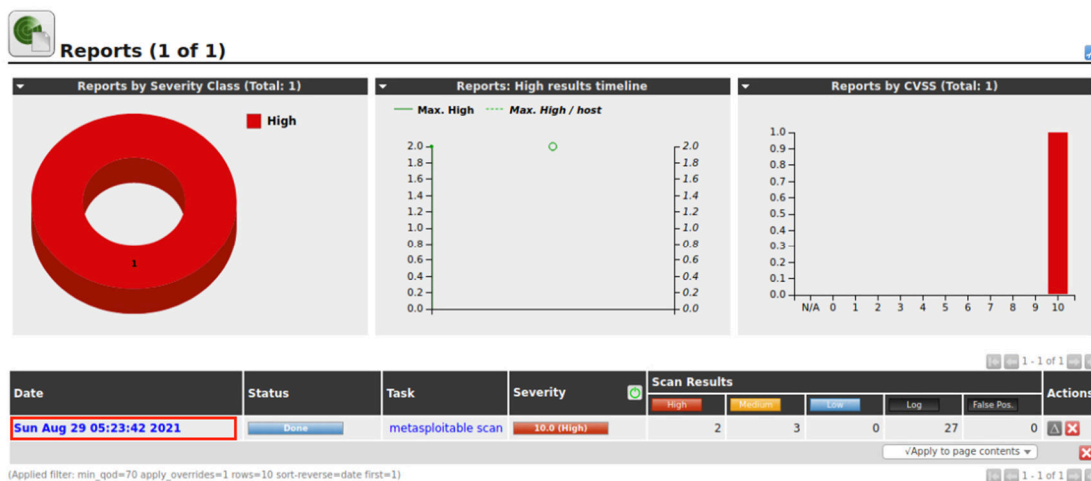
- To the upper-right corner, click the **arrow** in the *No auto-refresh* field to change the auto-refresh rate.




- When the scan is complete, under the **Scans** menu, click the **Reports** button.



- On the **Reports** page, click the data of your report to view and explore the vulnerabilities.













5. On the *Reports* page, you can click any of the entries in the *Vulnerability* column to check the detail.




Report: Results (5 of 38)

ID: 5c48d9e4-7239-4ba7-8e94-b1fdb4e3c93b
 Modified: Sun Aug 29 05:35:36 2021
 Created: Sun Aug 29 05:24:00 2021
 Owner: admin



Vulnerability	Severity	QoD	Host	Location	Actions
OpenVAS / Greenbone Vulnerability Manager Default Credentials	10.0 (High)	100%	127.0.0.1 (localhost)	9390/tcp	 
Redis Server No Password	7.5 (High)	100%	127.0.0.1 (localhost)	6379/tcp	 
SSL/TLS: Certificate Expired	5.0 (Medium)	99%	127.0.0.1 (localhost)	443/tcp	 
SSL/TLS: Certificate Expired	5.0 (Medium)	99%	127.0.0.1 (localhost)	9390/tcp	 
Check if Mailserver answer to VRFY and EXPN requests	5.0 (Medium)	99%	127.0.0.1 (localhost)	25/tcp	 

(Applied filter: autotp=0 apply_overrides=1 notes=1 overrides=1 result_hosts_only=1 first=1 rows=100 sort-reverse=severity levels=hml min_qod=70)



Result: OpenVAS / Greenbone Vulnerability Manager Default Credentials


ID: 7d3e2bf0-443b-40f6-8053-ef4ee8d97277
 Created: Sun Aug 29 05:30:32 2021
 Modified: Sun Aug 29 05:30:32 2021
 Owner: admin

Vulnerability	Severity	QoD	Host	Location	Actions
OpenVAS / Greenbone Vulnerability Manager Default Credentials	10.0 (High)	100%	127.0.0.1	9390/tcp	 

Summary
 The remote OpenVAS / Greenbone Vulnerability Manager is installed/configured in a way that it has account(s) with default passwords enabled.

Vulnerability Detection Result
 It was possible to login using the following credentials (username:password:role):
 admin:admin:Admin

Impact
 This issue may be exploited by a remote attacker to gain access to sensitive information or modify system configuration.

Solution
Solution type:  Workaround
 Change the password of the mentioned account(s).

Vulnerability Insight
 It was possible to login with default credentials: admin/admin, sadmin/changeme, observer/observer or admin/openvas.

Vulnerability Detection Method
 Try to login with default credentials via the OMP/GMP protocol.
 Details: [OpenVAS / Greenbone Vulnerability Manager Default Credentials \(OID: 1.3.6.1.4.1.25623.1.0.108554\)](#)
 Version used: \$Revision: 13944 \$

Product Detection Result
 Product: [cpe:/a:openvas:openvas_manager:7.0](#)
 Method: [OpenVAS / Greenbone Vulnerability Manager Detection \(OID: 1.3.6.1.4.1.25623.1.0.103825\)](#)
 Log: [View details of product detection](#)

6. The lab is now complete; you may end the reservation.