



## **CLOUD SECURITY FUNDAMENTALS V2**

### **Lab 6: Container Vulnerability Scanning**

**Document Version: 2022-12-22**

Copyright © 2022 Network Development Group, Inc.  
[www.netdevgroup.com](http://www.netdevgroup.com)

NETLAB+ is a registered trademark of Network Development Group, Inc.

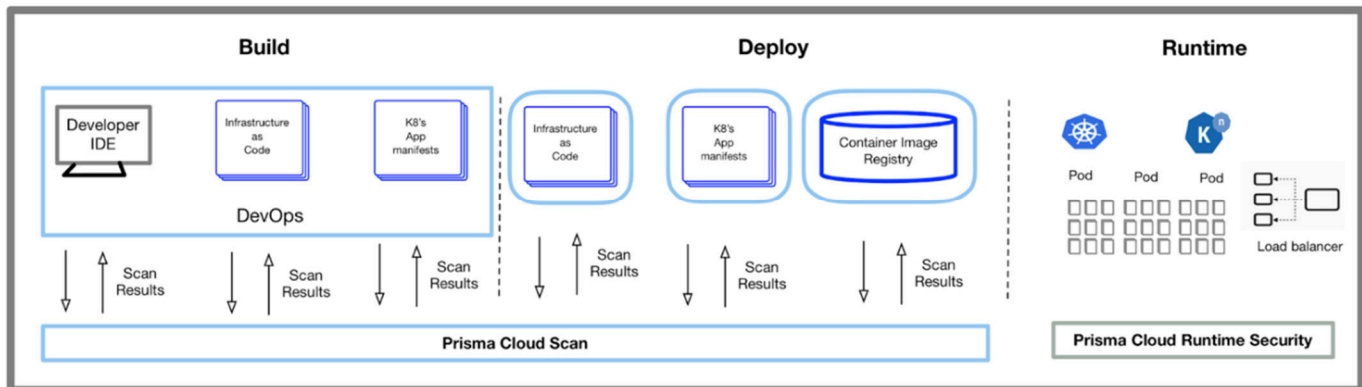
Palo Alto Networks and the Palo Alto Networks logo are trademarks or registered trademarks of Palo Alto Networks, Inc.

## Contents

Introduction .....	3
Objective .....	3
Lab Topology .....	4
Lab Settings .....	5
1 Container Vulnerability Scanning .....	6
1.0 Load Lab Configuration .....	6
1.1 Create a DVWA and Metasploit Container Targets on the DMZ Server .....	11
1.2 Use Docker Compose to Create an OpenVas Scanning Container.....	17
1.3 Conduct a Vulnerability Scan of the DMZ Server using an OpenVas Container	20
1.4 Prisma Cloud Vulnerability Scanner. ....	27

## Introduction

In this lab, you will create a DVWA container target on the DMZ server for vulnerability scanning. You will scan the DMZ server using a container to generate a report based on the vulnerabilities found on the Metasploit container running on the DMZ server. The Metasploit container will be a target for the OpenVAS vulnerability scanner container running on the client computer. You will also view Prisma Cloud reports that will help mitigate risks and secure workloads in a hybrid / multicloud environment.

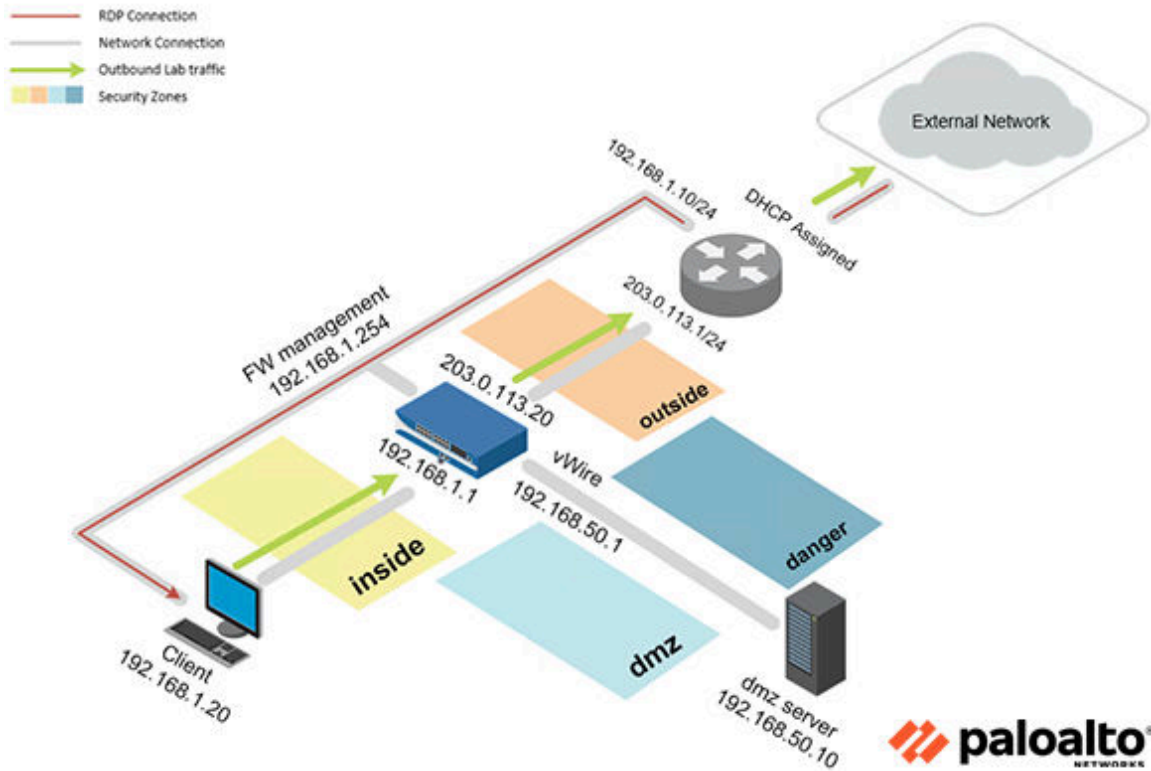


## Objective

In this lab, you will perform the following tasks:

- Create a docker DVWA container target on the DMZ server for vulnerability scanning
- Create an OpenVas container using docker-compose
- Scan the DMZ server using an OpenVas container and generate a report
- View the OpenVas report
- View Prisma Cloud scanning reports

## 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.

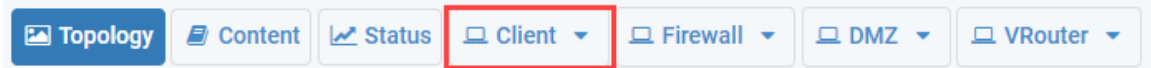
Virtual Machine	IP Address	Account (if needed)	Password (if needed)
Client	192.168.1.20	lab-user	Pal0Alt0!
DMZ	192.168.50.10	root	Pal0Alt0!
Firewall	192.168.1.254	admin	Pal0Alt0!

## 1 Container Vulnerability Scanning

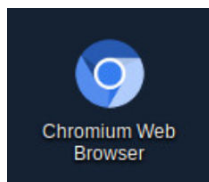
### 1.0 Load Lab Configuration

In this section, you will load the Firewall configuration file.

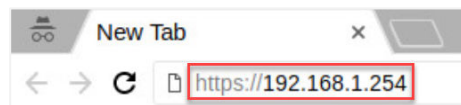
1. Click on the **Client** tab to access the Client PC.



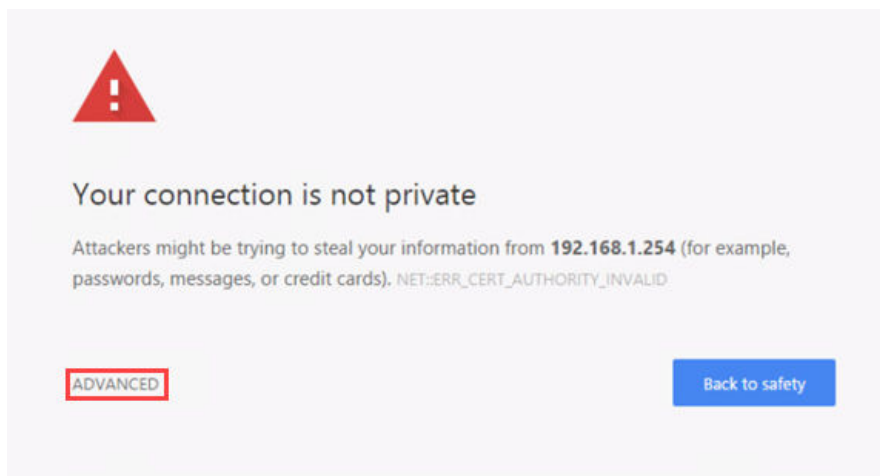
2. Log in to the Client PC as username `lab-user`, password `Pa10Alt0!`.
3. Double-click the **Chromium Web Browser** icon located on the Desktop.



4. In the *Chromium* address field, type `https://192.168.1.254` and press **Enter**.

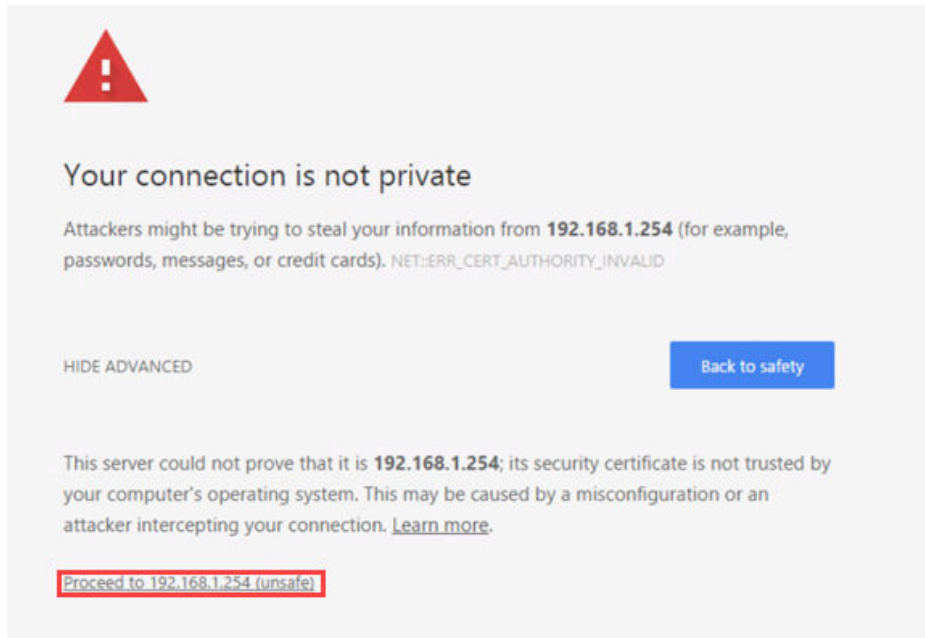


5. You will see a “Your connection is not private” message. Click on the **ADVANCED** link.



If you experience the “Unable to connect” or “502 Bad Gateway” message while attempting to connect to the specified IP above, please wait an additional 1-3 minutes for the Firewall to fully initialize. Refresh the page to continue.

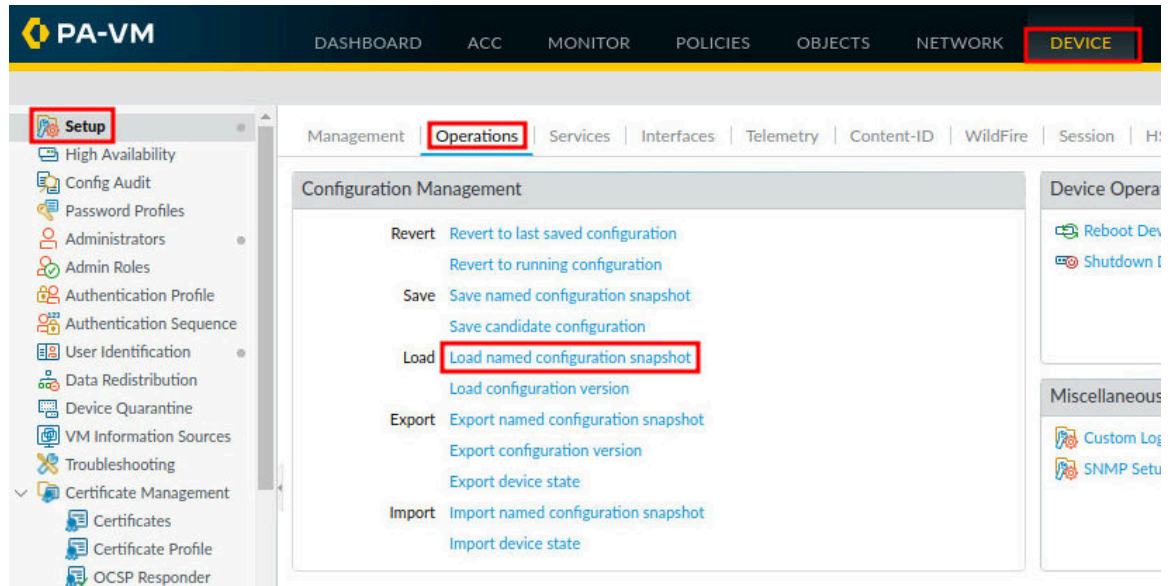
6. Click on **Proceed to 192.168.1.254 (unsafe)**.



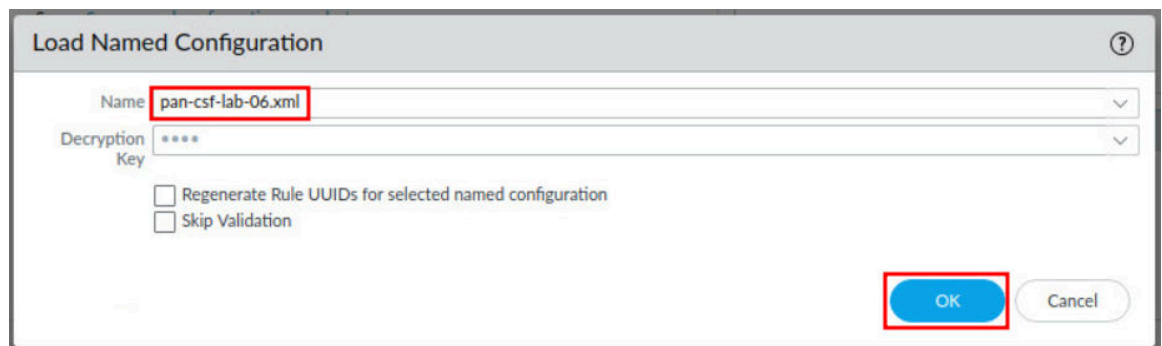
7. Log in to the Firewall web interface as username admin, password Pal0Alt0!.



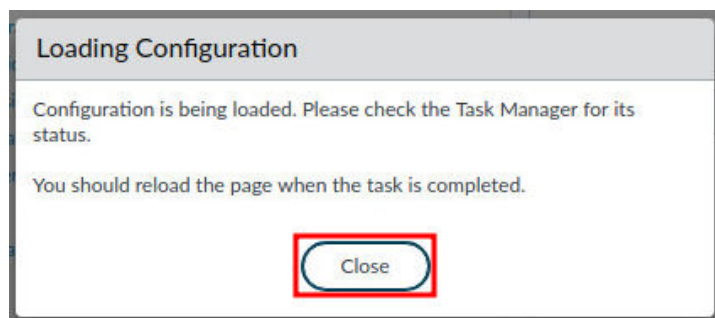
- In the web interface, navigate to **Device > Setup > Operations** and click on **Load named configuration snapshot** underneath the *Configuration Management* section.



- In the *Load Named Configuration* window, select **pan-csf-lab-06.xml** from the *Name* dropdown box and click **OK**.

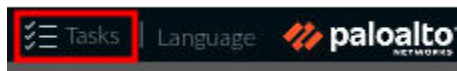


- In the Loading Configuration window, a message will show *Configuration is being loaded. Please check the Task Manager for its status. You should reload the page when the task is completed.* Click **Close** to continue.

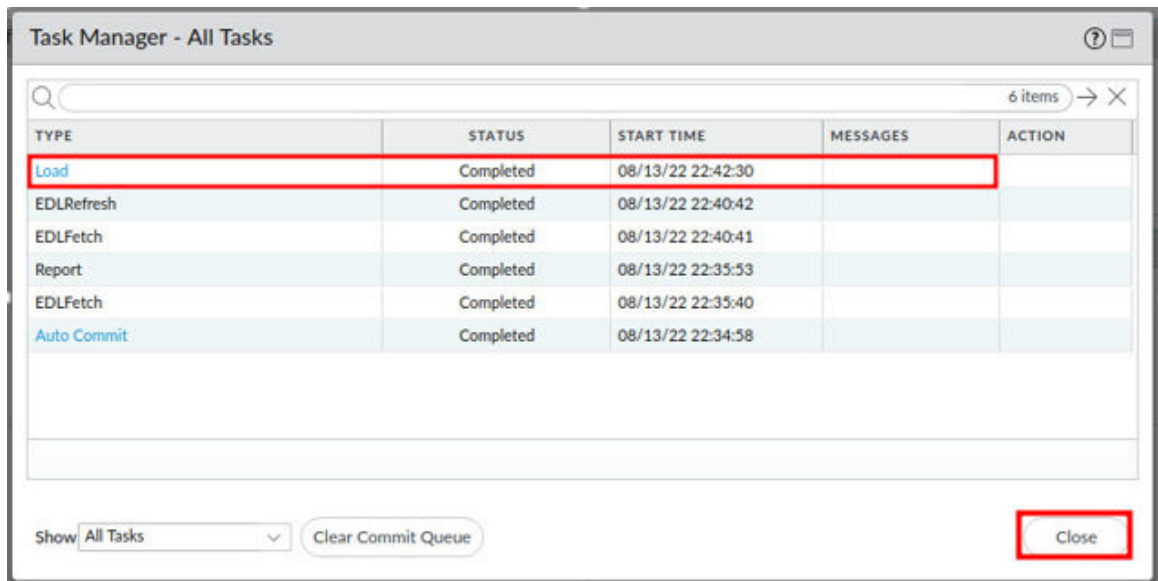




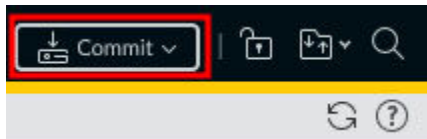
11. Click the **Tasks** icon located at the bottom-right of the web interface.



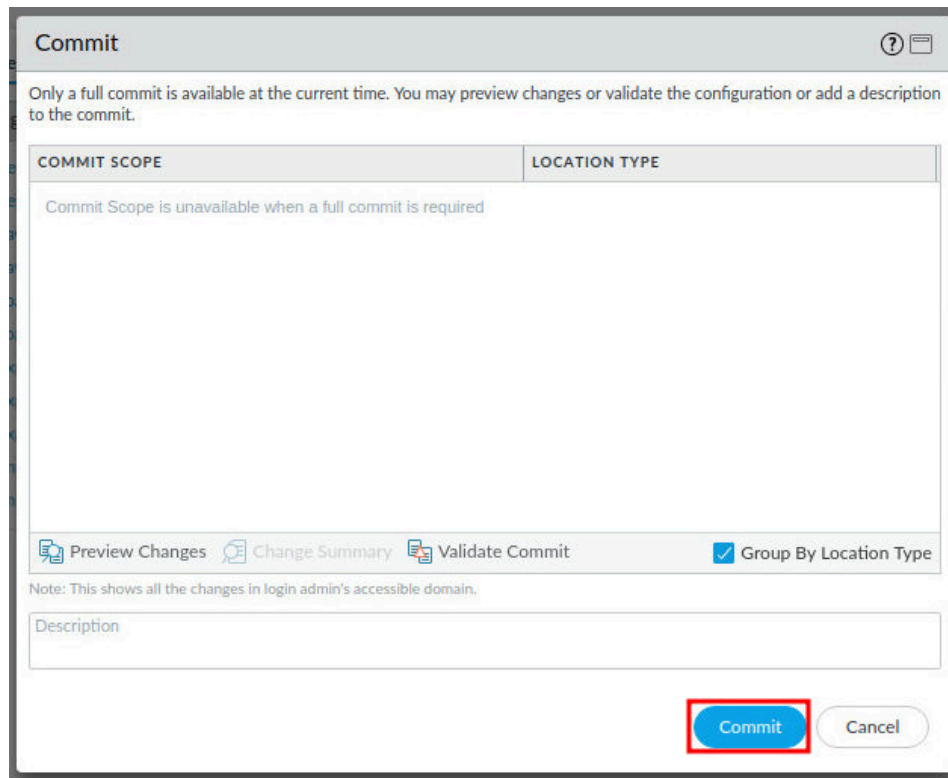
12. In the *Task Manager – All Tasks* window, verify the *Load* type has successfully completed. Click **Close**.



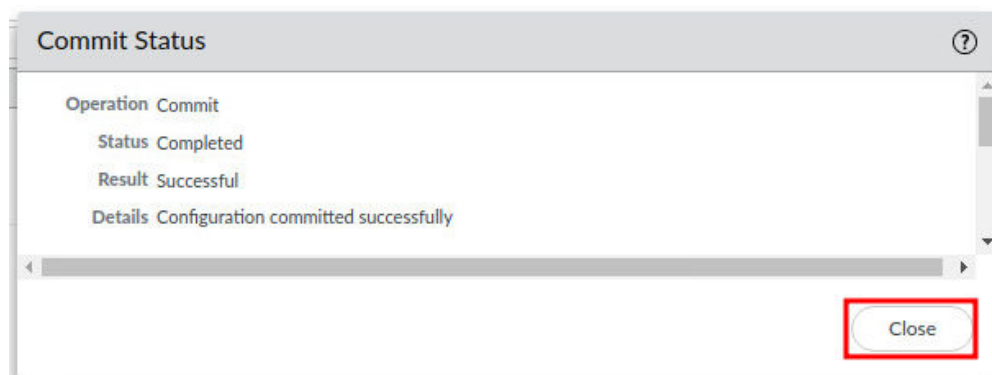
13. Click the **Commit** link located at the top-right of the web interface.



14. In the *Commit* window, click **Commit** to proceed with committing the changes.



15. When the commit operation successfully completes, click **Close** to continue.



The commit process takes changes made to the Firewall and copies them to the running configuration, which will activate all configuration changes since the last commit.

## 1.1 Create a DVWA and Metasploit Container Targets on the DMZ Server

In this section, you will create a DVWA container. You will use a metasploitable container to serve as a target for exploit testing to improve your defensive skills.

1. Launch **Xfce** Terminal in the lower-left of the student *Desktop*.



2. SSH to the *DMZ* by typing the command below. Use **Pa10Alt0!** for the password. If prompted, enter **yes** to continue connecting. Press **Enter**.

```
C:\home\lab-user> ssh root@192.168.50.10
```



3. Launch a metasploitable container on the DMZ server by typing the command below.

```
[root@pod-dmz ~]# docker run -ditP --name metasploit icarossio/metasploitable2
```

```
[root@pod-dmz ~]# docker run -ditP --name metasploit icarossio/metasploitable2  
679d33f9d1ac3ed0237ca928f7f7e6b645a4d62e1da306aa5eee62c11a81ada3  
[root@pod-dmz ~]#
```



This output will display all the DMZ server ports mapped to the metasploitable container. Metasploitable is a vulnerable Linux server intended to serve as a target for exploit testing to improve your defensive skills.

4. Display all the DMZ server ports mapped to the metasploitable container by typing the command below.

```
[root@pod-dmz ~]# docker port metasploit
```

```
[root@pod-dmz ~]# docker port metasploit
3306/tcp -> 0.0.0.0:32774
3632/tcp -> 0.0.0.0:32773
512/tcp -> 0.0.0.0:32779
6667/tcp -> 0.0.0.0:32769
80/tcp -> 0.0.0.0:32783
139/tcp -> 0.0.0.0:32781
21/tcp -> 0.0.0.0:32787
2121/tcp -> 0.0.0.0:32775
445/tcp -> 0.0.0.0:32780
513/tcp -> 0.0.0.0:32778
514/tcp -> 0.0.0.0:32777
5432/tcp -> 0.0.0.0:32772
6000/tcp -> 0.0.0.0:32770
111/tcp -> 0.0.0.0:32782
1524/tcp -> 0.0.0.0:32776
22/tcp -> 0.0.0.0:32786
23/tcp -> 0.0.0.0:32785
25/tcp -> 0.0.0.0:32784
5900/tcp -> 0.0.0.0:32771
8009/tcp -> 0.0.0.0:32768
[root@pod-dmz ~]#
```

5. Launch a DVWA container by typing the command below.

```
[root@pod-dmz ~]# docker run -d --name csf-dvwa -p 8080:80
vulnerables/web-dvwa
```

```
[root@pod-dmz ~]# docker run -d --name csf-dvwa -p 8080:80 vulnerables/web-dvwa
cdc8c6021e6bb144ad66e6f3a3a906792ef734b940941eba1c55514d88a629ca
[root@pod-dmz ~]#
```



This command will launch a container using container image `vulnerables/web-dvwa` and bridge the DMZ server's host port 8080 to the running container's port 80 which will allow you to access the DVWA container's webpage from your client.

- View the DVWA container by typing the command below.

```
[root@pod-dmz ~]# docker ps
```

```
[root@pod-dmz ~]# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
cdc8c6021e6b	vulnerables/web-dvwa	"/main.sh"	About a minute ago	Up About a minute	0.0.0.0:8080->80/tcp	
679d33f9d1ac	icarossio/metasploitable2	"/bin/sh -c '/bin/se..."	7 minutes ago	Up 7 minutes	0.0.0.0:32787->21/tcp, 0.0.0.0:32786->22/tcp, 0.0.0.0:32785->23/tcp, 0.0.0.0:32784->25/tcp, 0.0.0.0:32783->80/tcp, 0.0.0.0:32782->111/tcp, 0.0.0.0:32781->139/tcp, 0.0.0.0:32780->445/tcp, 0.0.0.0:32779->512/tcp, 0.0.0.0:32778->513/tcp, 0.0.0.0:32777->514/tcp, 0.0.0.0:32776->1524/tcp, 0.0.0.0:32775->2121/tcp, 0.0.0.0:32774->3306/tcp, 0.0.0.0:32773->3632/tcp, 0.0.0.0:32772->5432/tcp, 0.0.0.0:32771->5900/tcp, 0.0.0.0:32770->6000/tcp, 0.0.0.0:32769->6667/tcp, 0.0.0.0:32768->8009/tcp	metasploit

```
[root@pod-dmz ~]#
```

**Please Note**

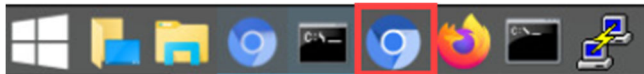
As you can see from the output of the command, container “**csf-dvwa**” using image **vulnerables/web-dvwa** with the DMZ server host port 8080 mapped to the container port 80 has been up for about a minute. Your results may vary on the uptime.

- Close the SSH connection to the DMZ server by typing **exit**.

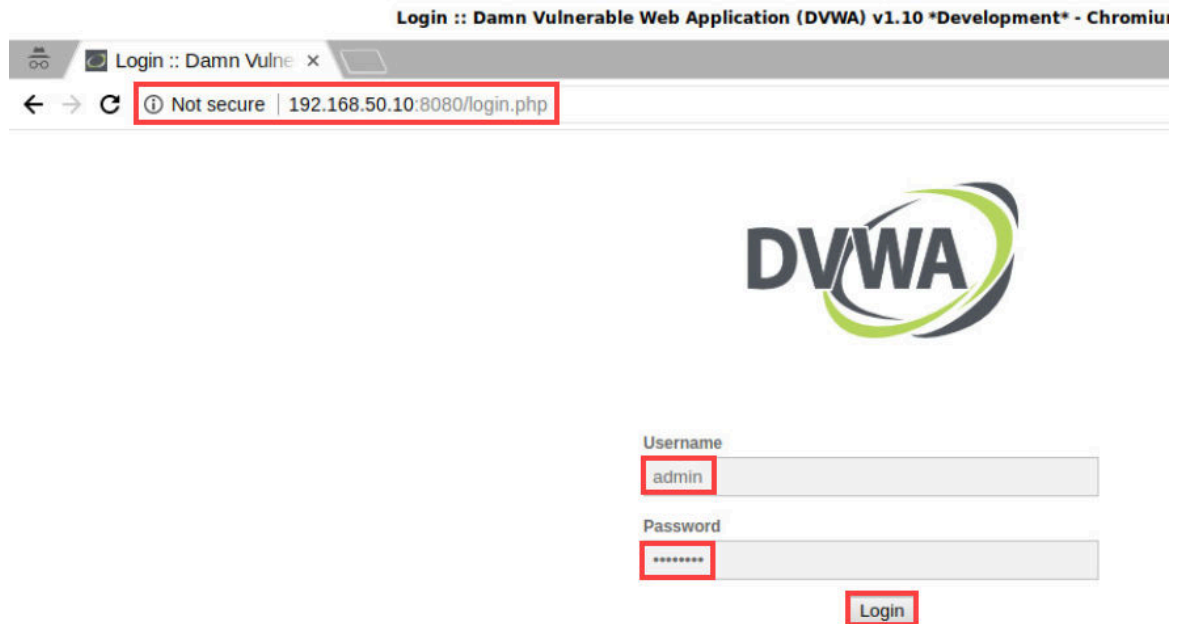
```
[root@pod-dmz ~]# exit
```

```
[root@pod-dmz ~]# exit
logout
Connection to 192.168.50.10 closed.
C:\home\lab-user>
```

- Launch a *Chromium Web Browser* by clicking the **Chromium** icon in the lower-left of the student *Desktop*.



9. Access the DVWA website by entering the URL `http://192.168.50.10:8080` and press **Enter**. On the DVWA webpage, enter `admin` for the *username* and `password` for the *password*. Click **Login**.



DVWA is a platform that can be used to learn how to conduct sql injections, cross site scripting and brute force attacks. There are multiple websites on the internet that will show you how to conduct web attacks against DVWA, here is one resource: <http://www.dvwa.co.uk/>. For this lab, you're going to use the DVWA website as a target for an OpenVas vulnerability scan.

10. On the *DVWA Database Setup* page, scroll to the bottom of the page and click the **Create / Reset Database** button.

PHP module mysql: **Installed**  
PHP module pdo\_mysql: **Installed**

MySQL username: **app**  
MySQL password: **\*\*\*\*\***  
MySQL database: **dvwa**  
MySQL host: **127.0.0.1**

reCAPTCHA key: **Missing**

[User: www-data] Writable folder /var/www/html/hackable/uploads/: **Yes**  
[User: www-data] Writable file /var/www/html/external/phpids/0.6/lib/IDS/tmp/phpids\_log.txt: **Yes**

[User: www-data] Writable folder /var/www/html/config: **Yes**  
**Status in red**, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your php.ini file and restart Apache.

`allow_url_fopen = On`  
`allow_url_include = On`

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

**Create / Reset Database**


First time using DVWA.  
Need to run 'setup.php'.

Damn Vulnerable Web Application (DVWA) v1.10 \*Development\*

11. On the *DVWA* webpage, log back in by entering *admin* for the *username* and *password* for the *password*. Click **Login**.

Login :: Damn Vulnerable Web Application (DVWA) v1.10 \*Development\* - Chrome

← → ↻ ⓘ Not secure | 192.168.50.10:8080/login.php



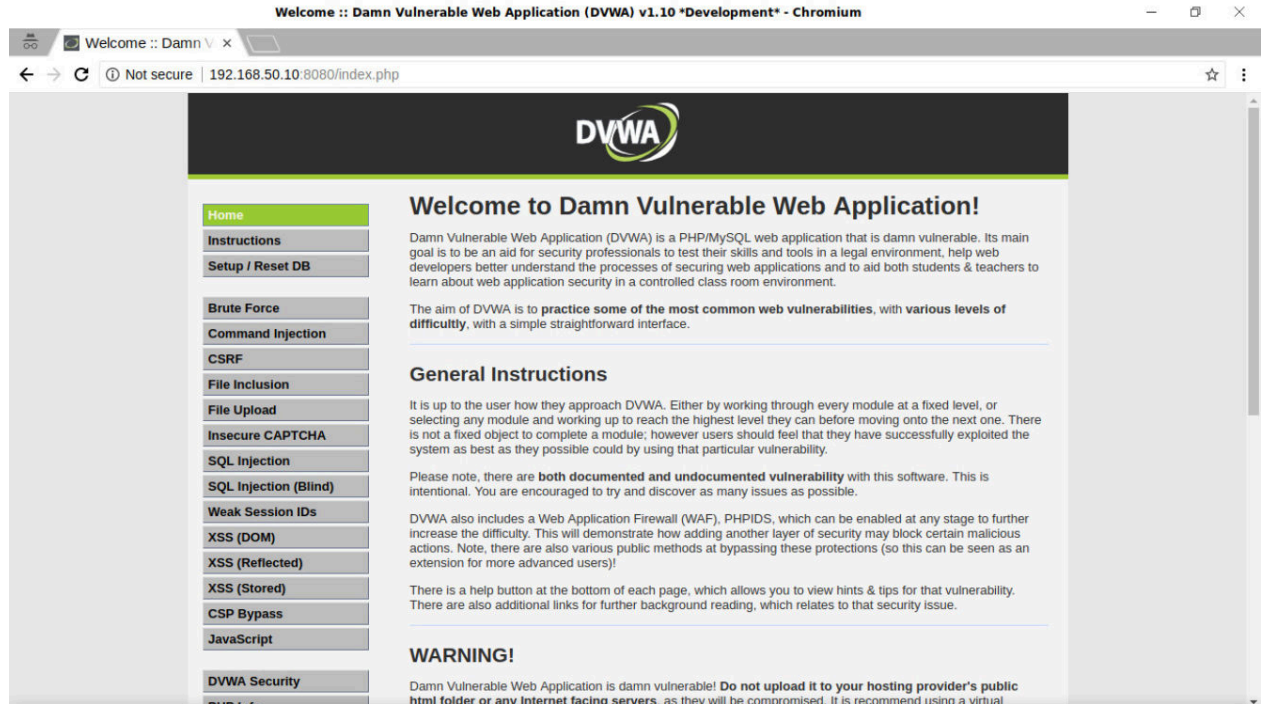
Username  
**admin**

Password  
**\*\*\*\*\***

**Login**



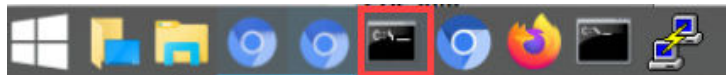
12. On the DVWA webpage, review the website after creating the database. Do not change any settings.



13. Close the *Welcome :: D....* window by clicking the **X** icon located at the top-right.



14. Change focus back to the *Terminal* by clicking on the **Terminal** icon located in the lower-left of the student *Desktop*. Leave the Terminal open and continue to the next task.





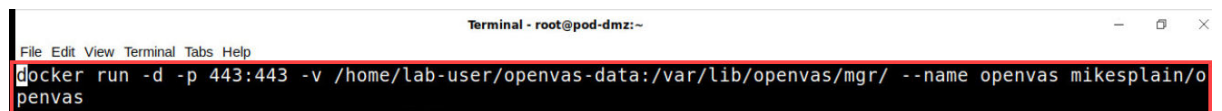
## 1.2 Use Docker Compose to Create an OpenVas Scanning Container

In this section, you will utilize Docker Compose to define and run multiple Docker container applications. You will use Docker Compose to run an OpenVas Docker container application to scan the DMZ server. Docker Compose uses a yaml declarative script to provide instructions to run container applications.

1. To view the docker run command that was used to create the docker-compose.yml, enter the command below.

```
C:\home\lab-user> vi docker-run-openvas
```

```
C:\home\lab-user> vi docker-run-openvas
```

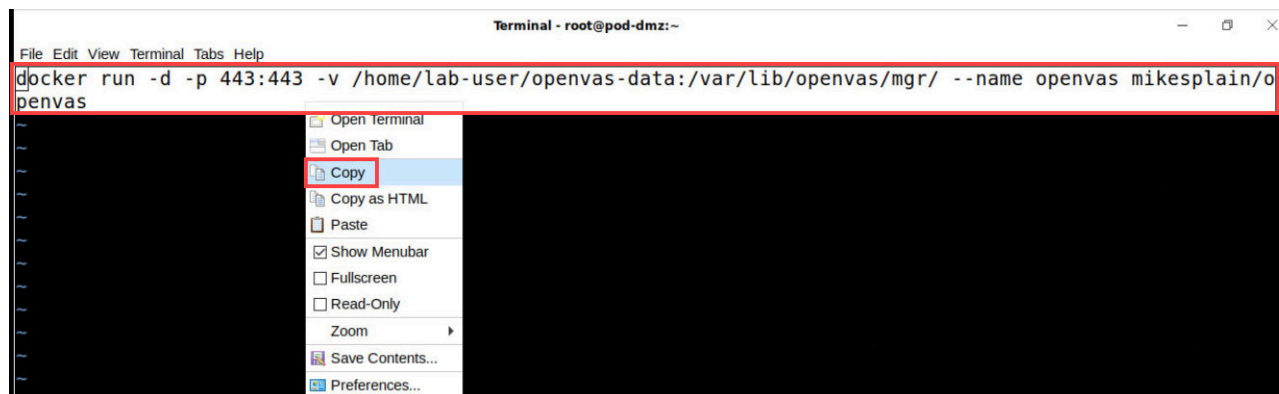


```
Terminal - root@pod-dmz:~  
File Edit View Terminal Tabs Help  
docker run -d -p 443:443 -v /home/lab-user/openvas-data:/var/lib/openvas/mgr/ --name openvas mikesplain/o  
penvas
```



This docker run command creates a container named openvas using the docker image mikesplain/openvas. The command maps the client port 443 to the container port 443 and maps the client's /home/lab-user/openvas-data directory/volume to the container's /var/lib/openvas/mgr directory/volume. By mapping a client volume to a container volume, the openvas container scan reports, etc., will persist after the container is stopped.

2. Highlight the **docker run** command from the docker-run-openvas file displayed on your terminal. After selecting the docker run command, right-click the run command and click **Copy**.

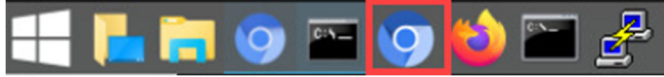


```
Terminal - root@pod-dmz:~  
File Edit View Terminal Tabs Help  
docker run -d -p 443:443 -v /home/lab-user/openvas-data:/var/lib/openvas/mgr/ --name openvas mikesplain/o  
penvas
```

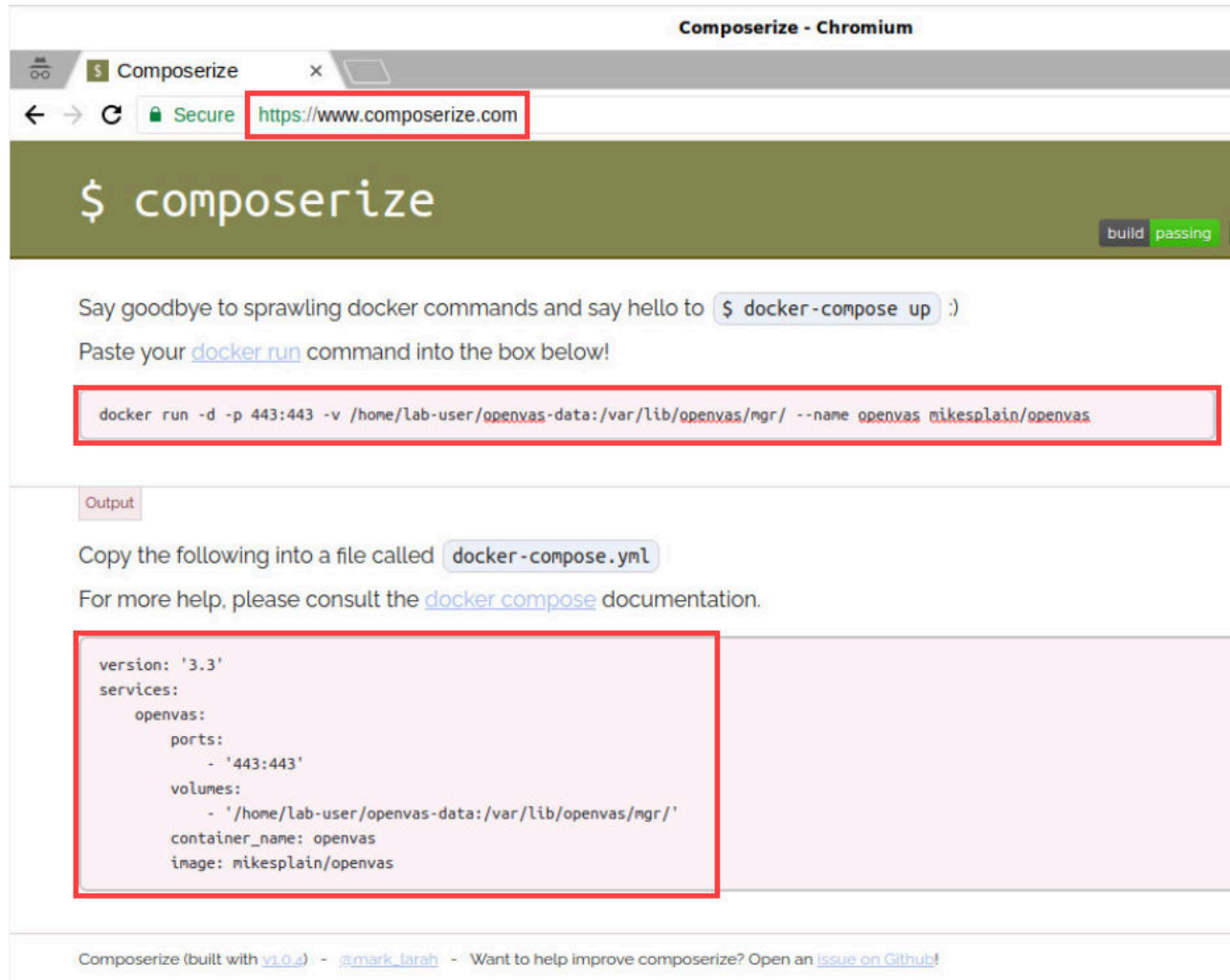
Context menu options:

- Open terminal
- Open Tab
- Copy
- Copy as HTML
- Paste
- Show Menubar
- Fullscreen
- Read-Only
- Zoom
- Save Contents...
- Preferences...

3. Open a new *Chromium* browser by clicking on the **Chromium** icon located in the lower-left of the student *Desktop*.



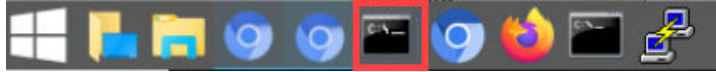
4. Enter the following URL `https://www.composerize.com` and click **Enter**. Paste the Docker run command from **step 2** into the website's docker run text box and view the output of the yaml file.



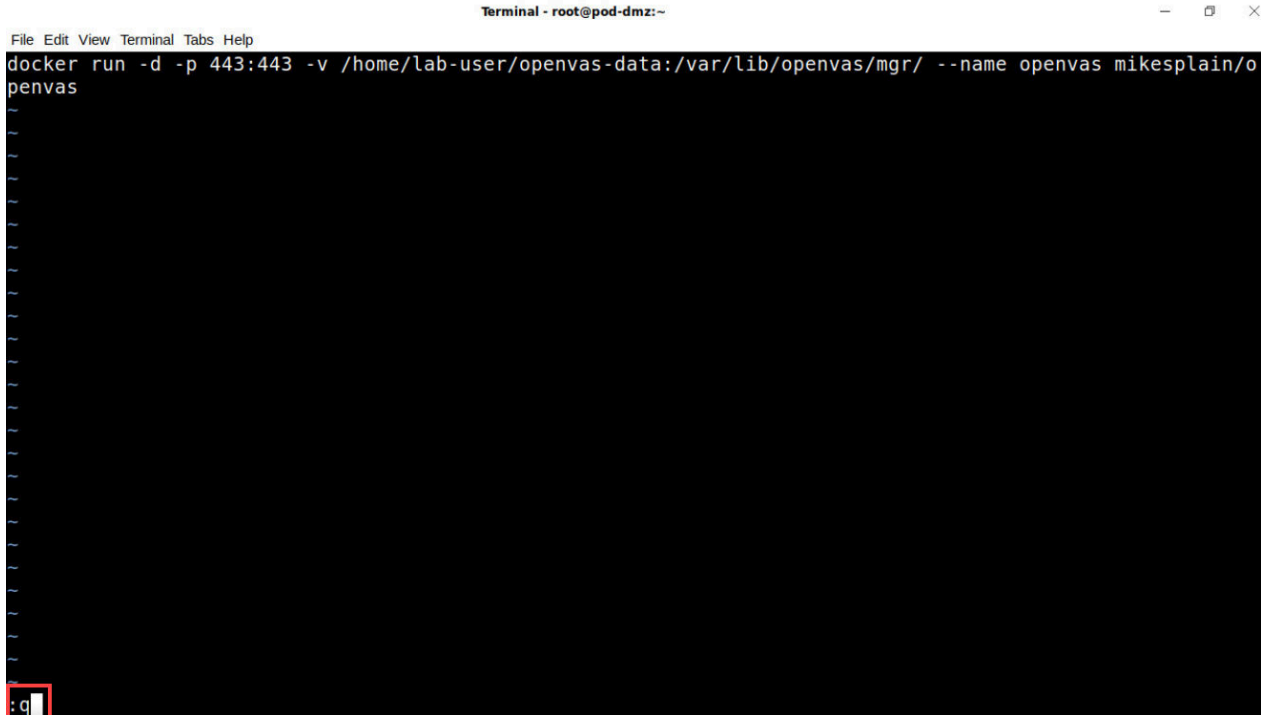
5. Close the *Composerize – Chromium* window by clicking the **X** icon located at the top-right.



6. Change focus back to the *terminal window* by clicking on the **Terminal** icon located in the lower-left of the student *Desktop*.



7. In the terminal window, type `:q` and click **Enter** to exit the *vi editor*.



8. To view the contents of the *docker-compose.yml* file, enter the command below. After viewing the contents of the *docker-compose.yml* file, type `:q` and press **Enter**.



- Run the OpenVas container by typing the command below.

```
C:\home\lab-user> docker-compose up -d
```

```
C:\home\lab-user> docker-compose up -d
Creating network "lab-user_default" with the default driver
Creating openvas ... done
C:\home\lab-user>
```

- Open a new *Chromium* browser by clicking on the **Chromium** icon located in the lower-left of the student *Desktop*.

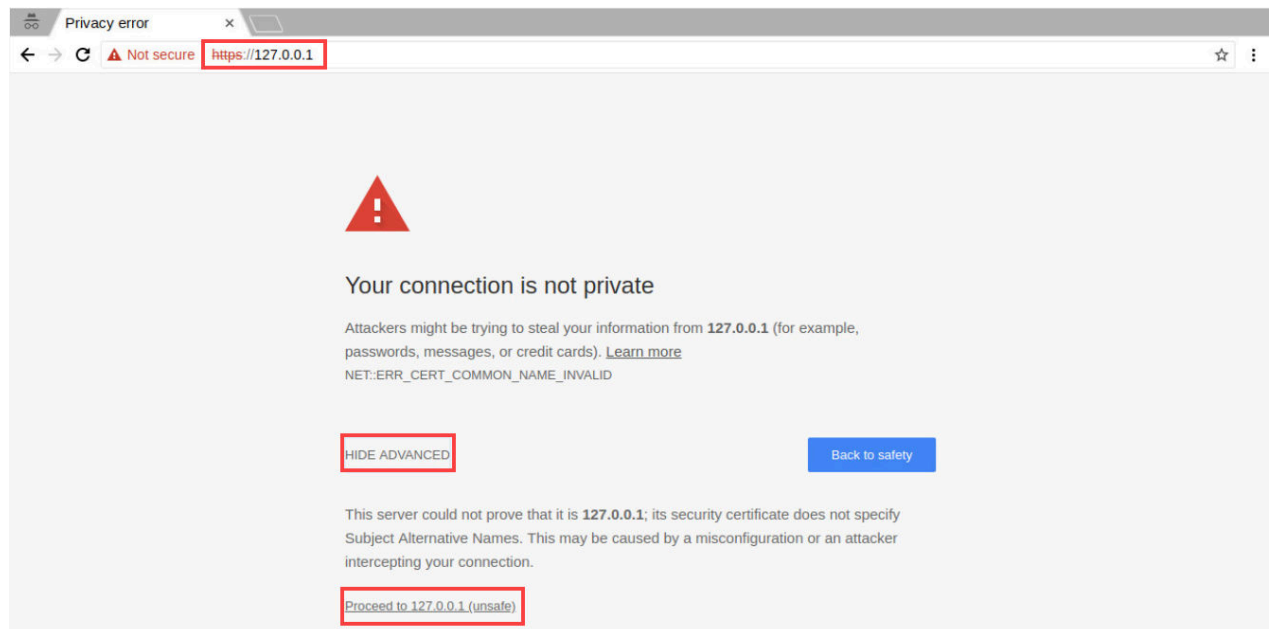


- Leave the new *Chromium* browser window open and continue to the next task.

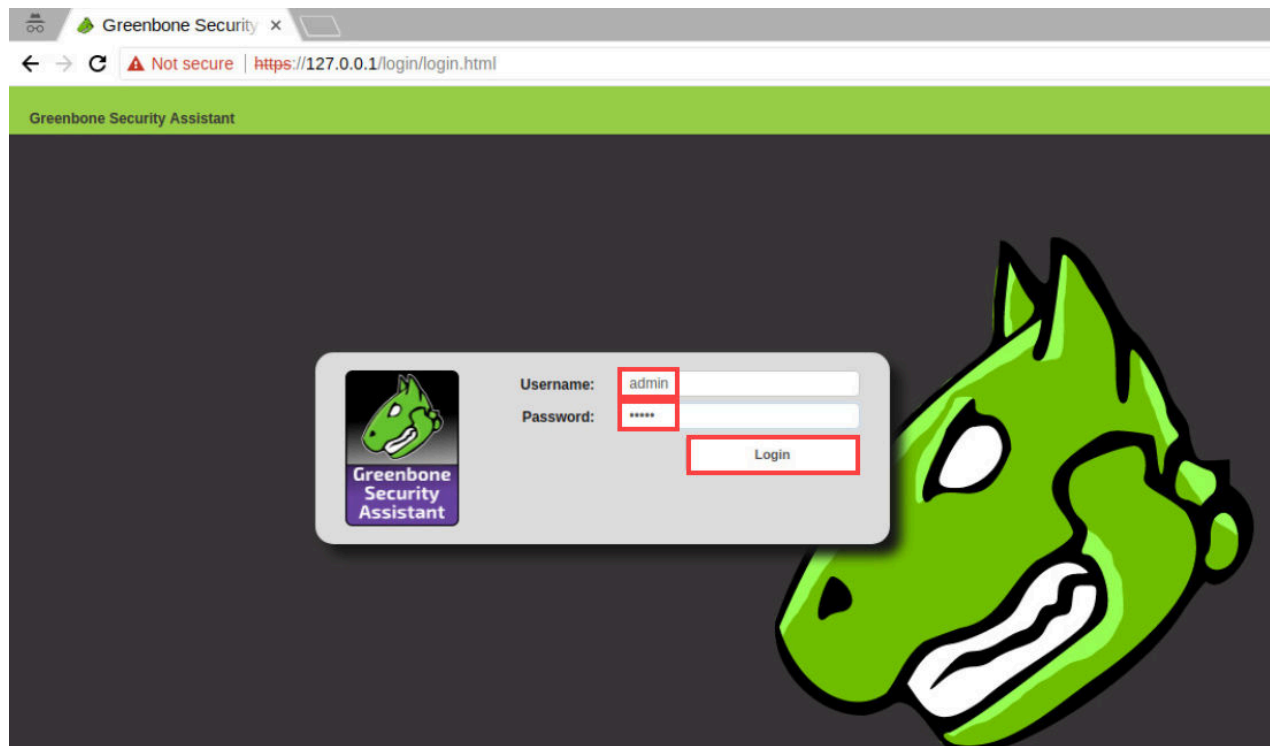
### 1.3 Conduct a Vulnerability Scan of the DMZ Server using an OpenVas Container

In this section, you will conduct a vulnerability scan of the DMZ server utilizing the OpenVas container.

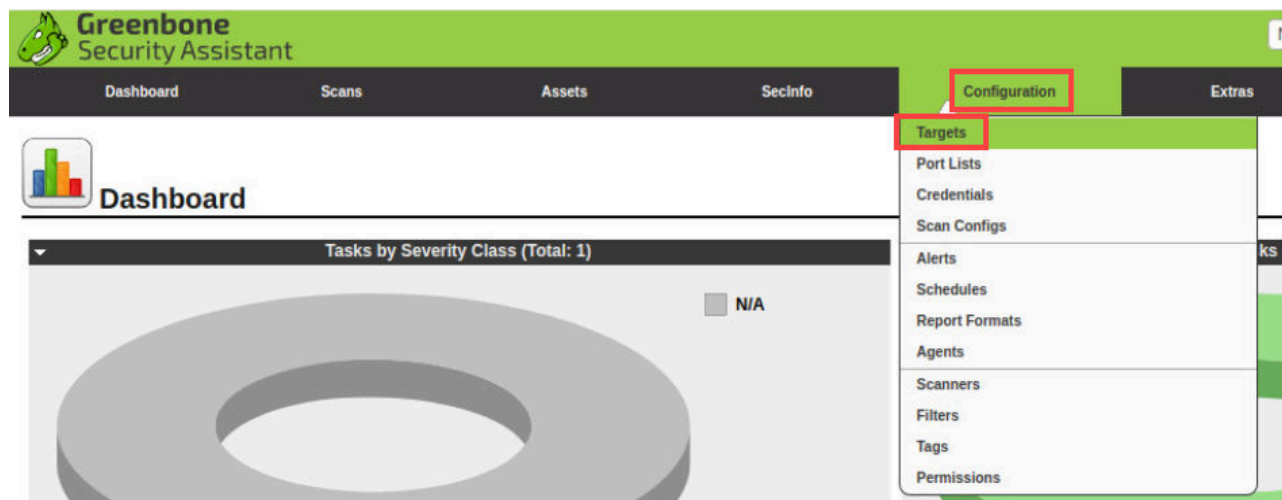
- Enter the following URL `https://127.0.0.1` and click **Enter**. Proceed through the *Your connection is not private*, by clicking **Advanced** and **Proceed to 127.0.0.1 (unsafe)**.



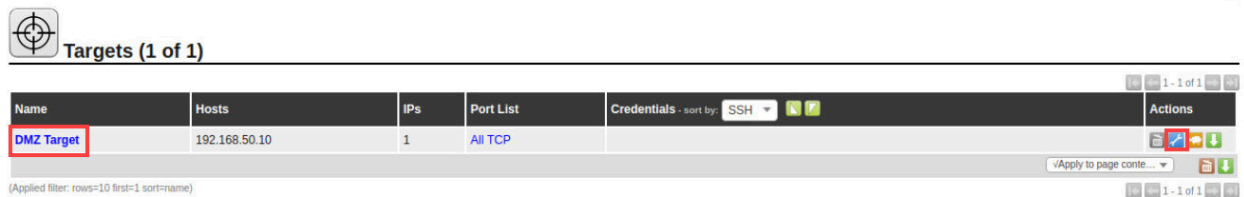
2. On the *OpenVas* website, enter *admin* for the *username* and *admin* for the *password*. Click **Login**.



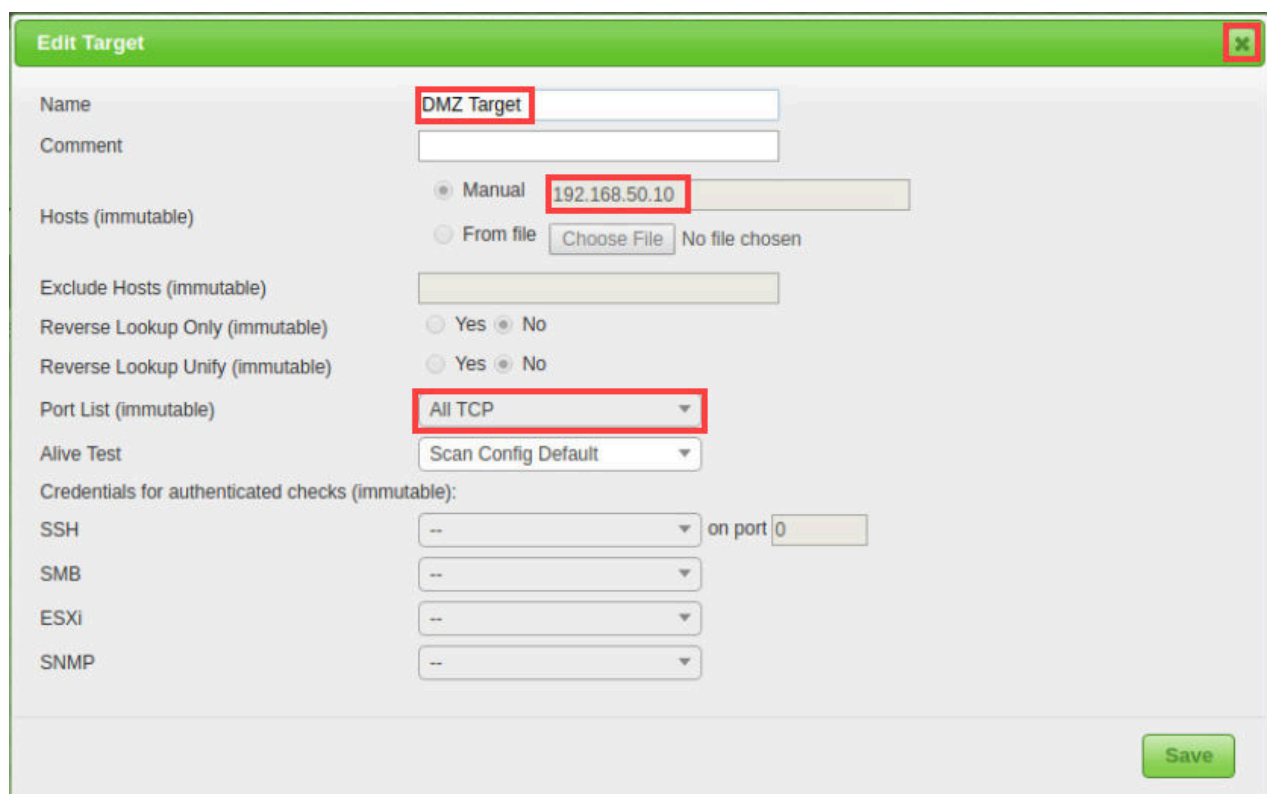
3. On the website's toolbar, click **Configuration > Targets** to access the configured targets.



- In the *Targets* webpage, click the **wrench** icon on the far-right of the **DMZ Target** to view the *configuration settings*.



- In the *Edit Target* dialogue box, observe the name **DMZ Target** and the *IP Address* of **192.168.50.10**. Lastly, notice that you are scanning all the **65,535 TCP ports**. After viewing the *Edit Target* details, click the **X** icon to close the window.



**Edit Target**

Name: DMZ Target

Comment:

Hosts (immutable): ☒ Manual 192.168.50.10 ☐ From file Choose File No file chosen

Exclude Hosts (immutable):

Reverse Lookup Only (immutable): ☐ Yes ☒ No

Reverse Lookup Unify (immutable): ☐ Yes ☒ No

Port List (immutable): All TCP

Alive Test: Scan Config Default

Credentials for authenticated checks (immutable):

SSH: -- on port 0

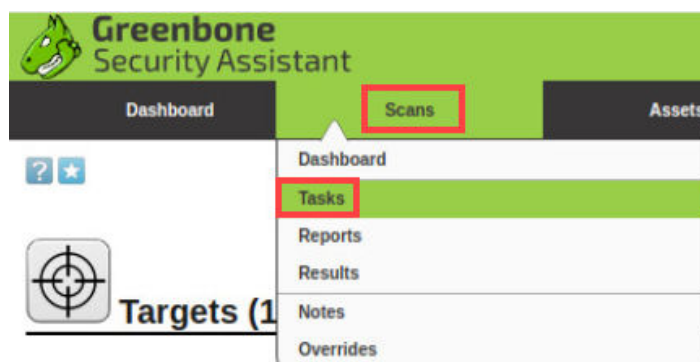
SMB: --

ESXi: --

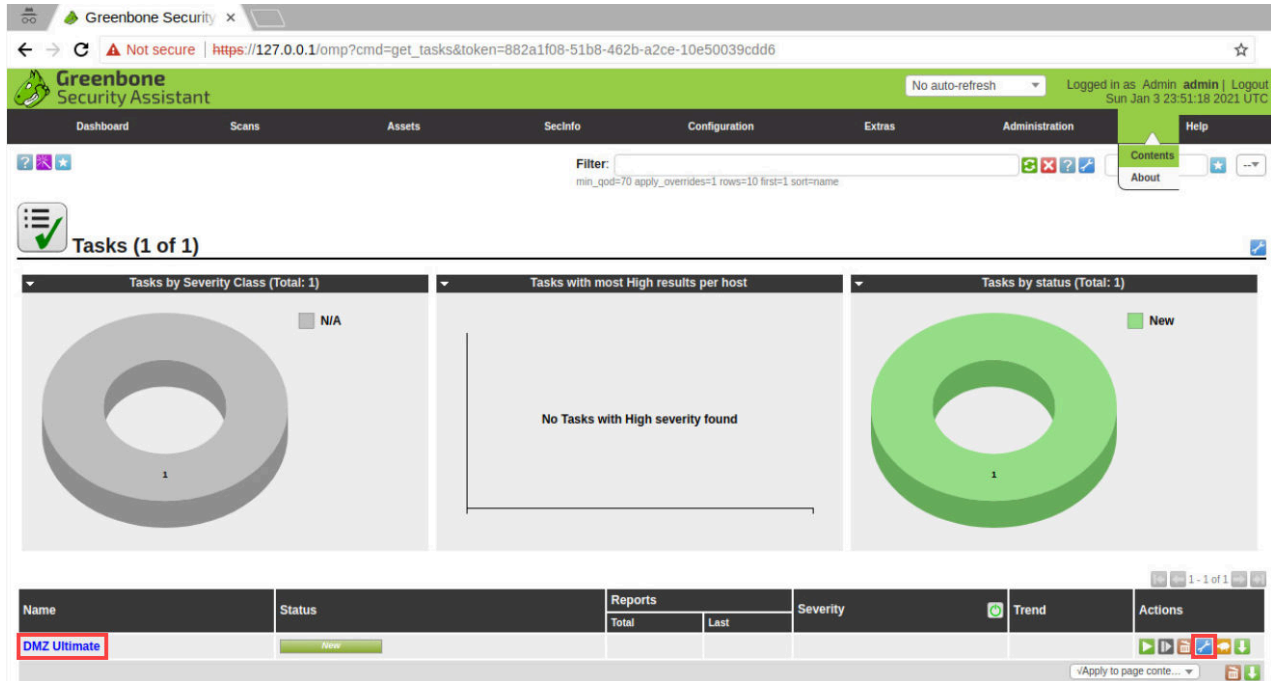
SNMP: --

Save

- On the *website's* toolbar, click **Scans > Tasks** to access the *Tasks*.



7. Scroll down to the bottom of the *task* window, click on the **wrench** icon for *DMZ Ultimate* and view the *configuration* settings.



Greenbone Security Assistant

Dashboard Scans Assets Secinfo Configuration Extras Administration Help

Filter: min\_qod=70 apply\_overrides=1 rows=10 first=1 sort=name

Tasks (1 of 1)

Tasks by Severity Class (Total: 1)

N/A

1

Tasks with most High results per host

No Tasks with High severity found

Tasks by status (Total: 1)

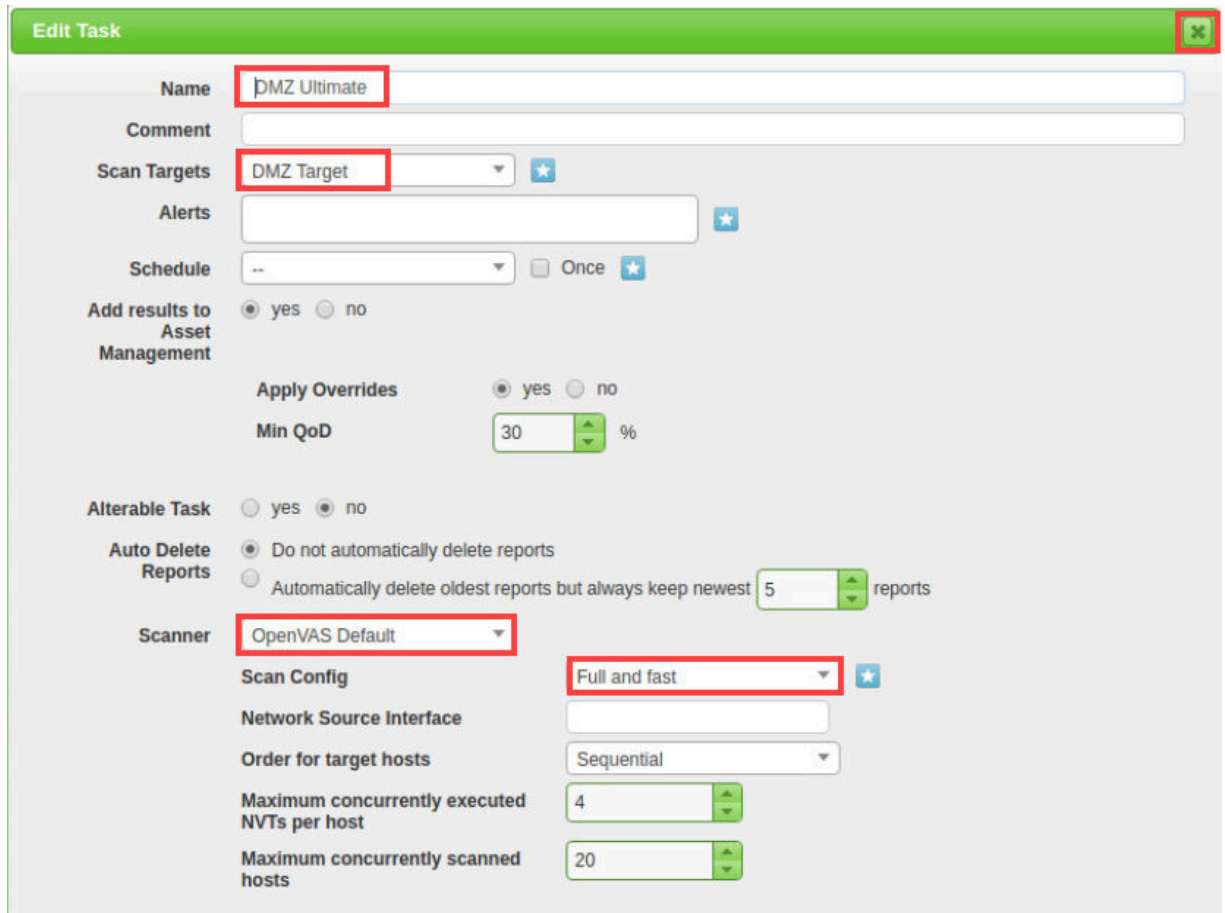
New

1

Name	Status	Reports	Severity	Trend	Actions
		Total	Last		
DMZ Ultimate	New				

Apply to page conte...

8. In the *Edit Task* dialogue box, note that the *scan target* is the **DMZ Target** that you just viewed. Note the scanner is the **OpenVas Default** scanner is using the **Full and fast** scan configuration. After viewing the task configurations, close the window by clicking the **X** icon.



**Edit Task** [X]

Name: **DMZ Ultimate**

Comment:

Scan Targets: **DMZ Target** [★]

Alerts: [★]

Schedule: -- [Once] [★]

Add results to Asset Management: ☒ yes ☐ no

Apply Overrides: ☒ yes ☐ no

Min QoD: 30 %

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:

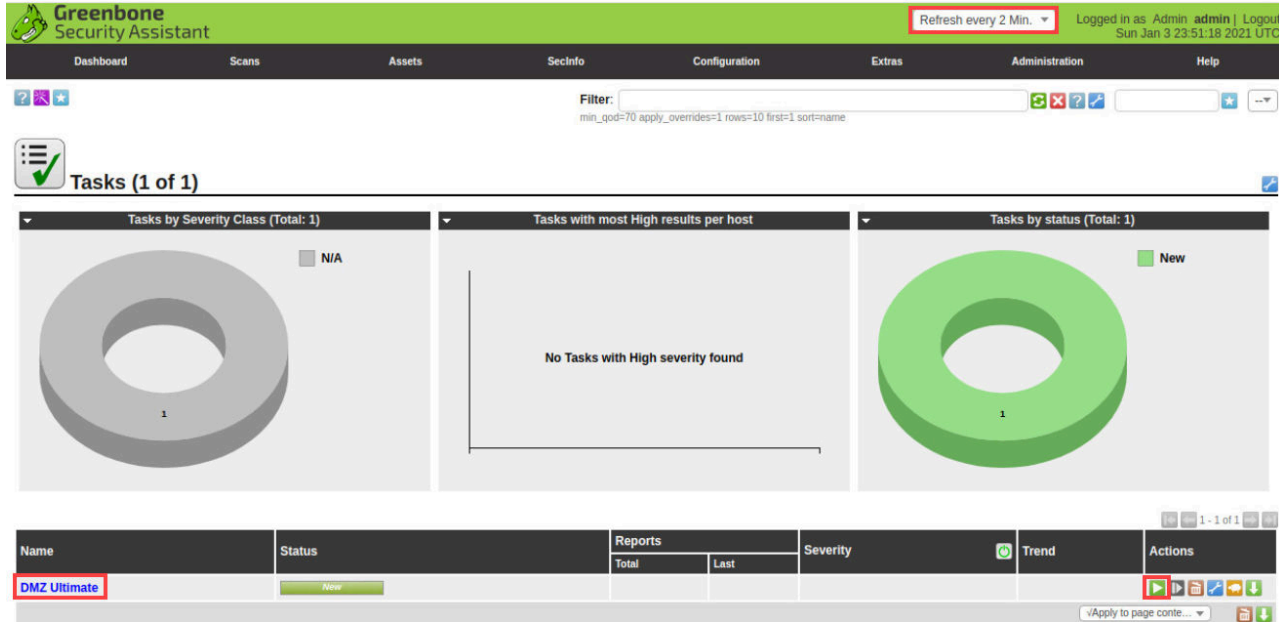
Order for target hosts: Sequential

Maximum concurrently executed NVTs per host: 4

Maximum concurrently scanned hosts: 20



9. On the **Scans > Tasks** webpage, select **Refresh every 2 min** at the top-right. Lastly, select the **Play** icon to the far-right of **DMZ Ultimate** to start a scan.

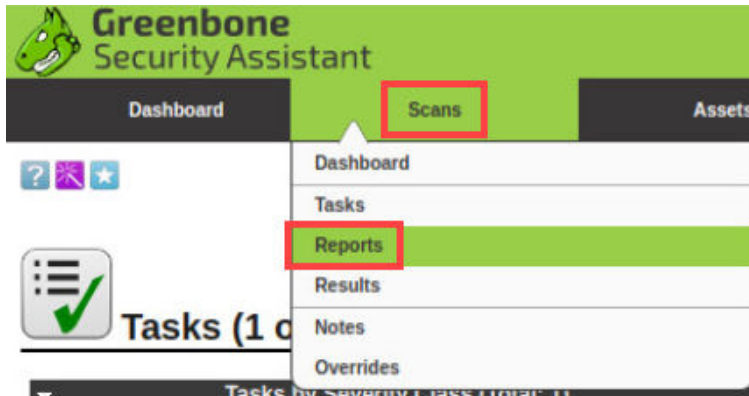


The screenshot shows the Greenbone Security Assistant interface. At the top, the 'Refresh every 2 Min' dropdown is highlighted in red. Below the navigation bar, the 'Tasks (1 of 1)' section is visible. The 'DMZ Ultimate' task is highlighted in red, and its 'Play' icon is also highlighted in red.

**Please Note**

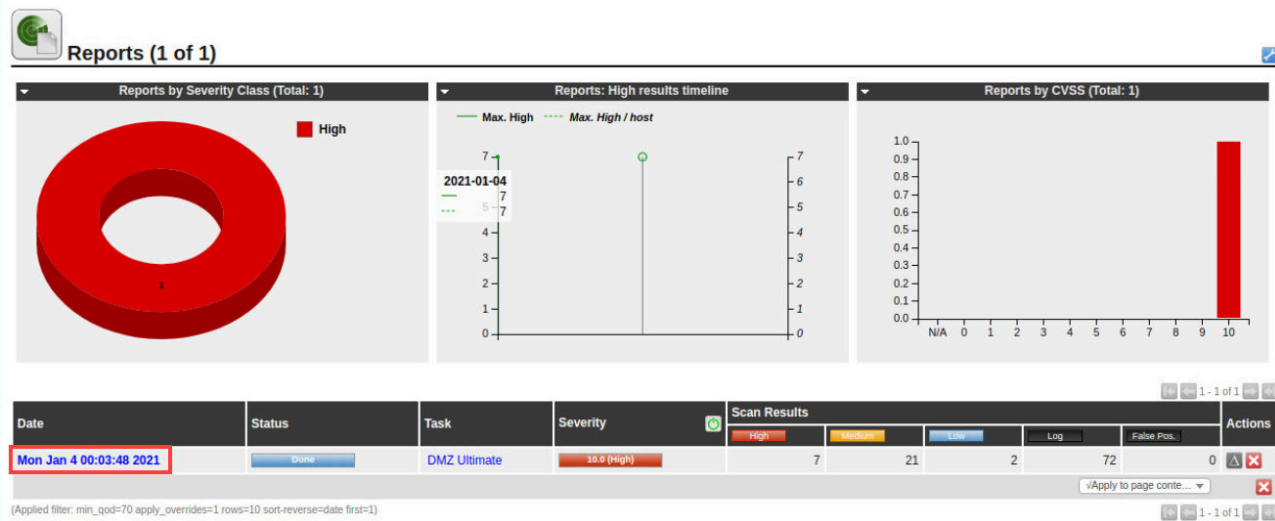
It will take up to 30 minutes to complete the scan. You may proceed to the next task of this lab while you are waiting for the vulnerability scan to complete. The Prisma Cloud section is informational and there are no lab activities.

10. After the scan is complete, select **Scans > Reports**.

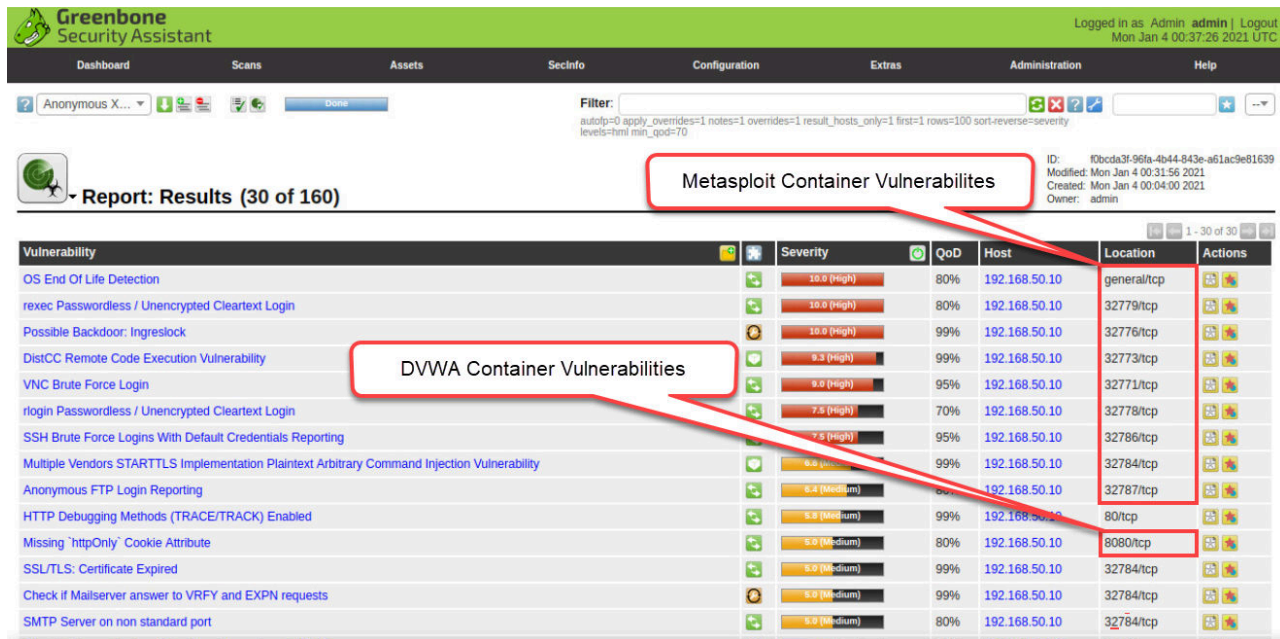


The screenshot shows the Greenbone Security Assistant interface. The 'Scans' menu item is highlighted in red, and the 'Reports' sub-menu item is also highlighted in red.

11. On the **Reports** webpage, click the date of your report to view and explore the vulnerabilities.



12. On the **Reports** webpage, view the report and notice the **vulnerabilities** found.



Vulnerability	Severity	QoD	Host	Location	Actions
OS End Of Life Detection	10.0 (high)	80%	192.168.50.10	general/tcp	
rexec Passwordless / Unencrypted Cleartext Login	10.0 (high)	80%	192.168.50.10	32779/tcp	
Possible Backdoor: Ingreslock	10.0 (high)	99%	192.168.50.10	32776/tcp	
DistCC Remote Code Execution Vulnerability	9.3 (high)	99%	192.168.50.10	32773/tcp	
VNC Brute Force Login	9.0 (high)	95%	192.168.50.10	32771/tcp	
rlogin Passwordless / Unencrypted Cleartext Login	7.5 (high)	70%	192.168.50.10	32778/tcp	
SSH Brute Force Logins With Default Credentials Reporting	7.5 (high)	95%	192.168.50.10	32786/tcp	
Multiple Vendors STARTTLS Implementation Plaintext Arbitrary Command Injection Vulnerability	6.8 (medium)	99%	192.168.50.10	32784/tcp	
Anonymous FTP Login Reporting	6.4 (medium)	80%	192.168.50.10	32787/tcp	
HTTP Debugging Methods (TRACE/TRACK) Enabled	5.9 (medium)	99%	192.168.50.10	80/tcp	
Missing 'httpOnly' Cookie Attribute	5.0 (medium)	80%	192.168.50.10	8080/tcp	
SSL/TLS: Certificate Expired	5.0 (medium)	99%	192.168.50.10	32784/tcp	
Check if Mailserver answer to VRFY and EXPN requests	5.0 (medium)	99%	192.168.50.10	32784/tcp	
SMTP Server on non standard port	5.0 (medium)	80%	192.168.50.10	32784/tcp	

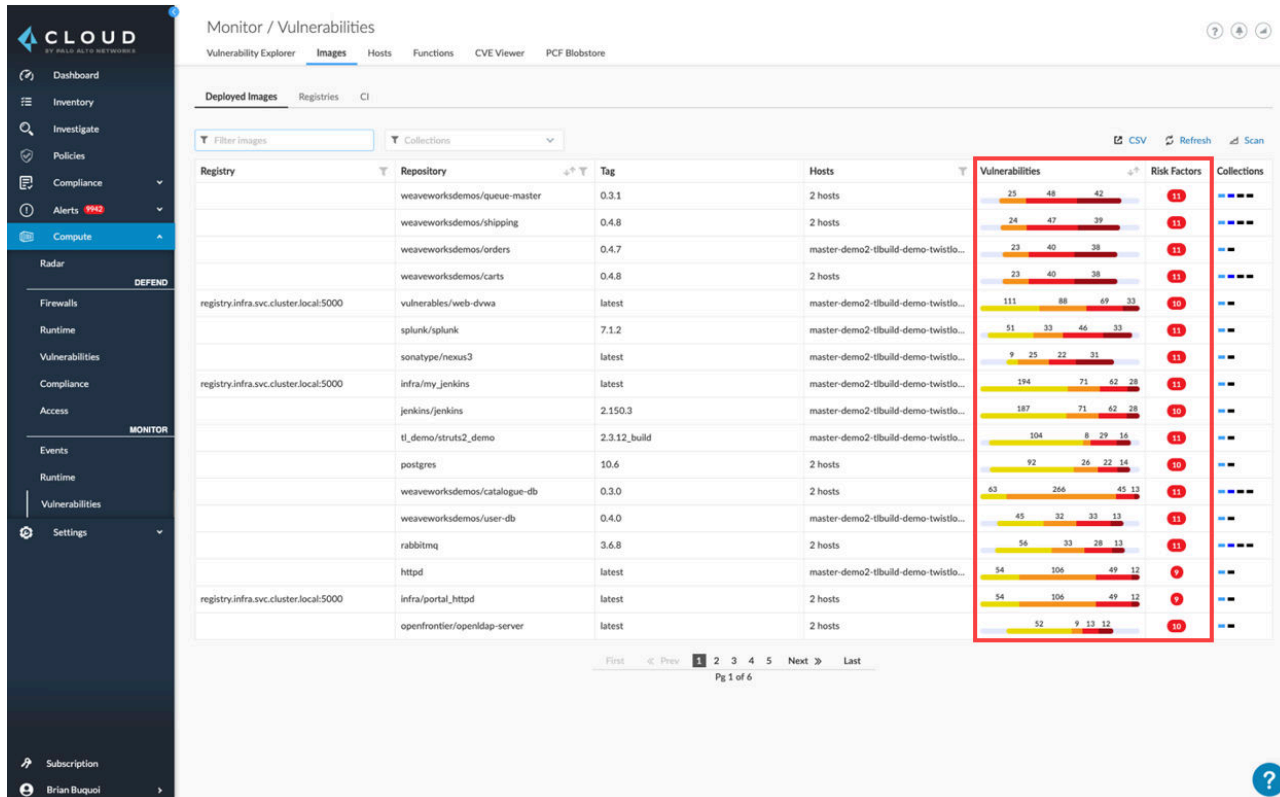
Please  
Note

Vulnerabilities for ports greater than 32000 are associated with the Metasploit container running on the DMZ server. Vulnerabilities for port 8080 are associated with the DVWA container running on the DMZ server.

## 1.4 Prisma Cloud Vulnerability Scanner.

In this section, you will get an overview of the Prisma Cloud Vulnerability Scanner. This section is informational only and does not have any lab activities.

1. Containers are one of the building blocks for Cloud Native Applications (CNAs). It usually takes several containers to run a single Cloud Native Application (CNA). Containers modularize code development by allowing separate development teams to work on applications and code running on separate containers that comprise a single CNA. Kubernetes provides an orchestration platform to run the individual containers as one seamless CNA. In a Dev/Ops environment, containers for CNAs are constantly being updated by developers to create a Continuous Integration/Continuous Deployment (CI/CD) pipeline.
2. In this lab, you performed a vulnerability scan against a running container, but it is also very important to scan container images during the development process before they are deployed. Prisma Cloud provides this capability by integrating directly into the CI/CD pipeline. Prisma Cloud also accelerates the deployment of new code running on containers by ensuring the container images are secured before they are deployed. The underlying goal of the risk score in Prisma Cloud is to create an actionable item that should be addressed and what urgency should be taken for the vulnerability.



The screenshot displays the Prisma Cloud Monitor / Vulnerabilities interface. The left sidebar contains navigation options: Dashboard, Inventory, Investigate, Policies, Compliance, Alerts (942), Compute, Radar, DEFEND (Firewalls, Runtime, Vulnerabilities, Compliance, Access), MONITOR (Events, Runtime, Vulnerabilities), and Settings. The main content area is titled 'Monitor / Vulnerabilities' and includes tabs for Vulnerability Explorer, Images, Hosts, Functions, CVE Viewer, and PCF Blobstore. The 'Images' tab is active, showing a table of deployed images. The table has columns for Registry, Repository, Tag, and Hosts. A detailed view of vulnerabilities is shown on the right, including a bar chart for each vulnerability and a table of risk factors.

Registry	Repository	Tag	Hosts	Vulnerabilities	Risk Factors	Collections
	weaveworksdemos/queue-master	0.3.1	2 hosts	25 48 42	11	---
	weaveworksdemos/shipping	0.4.8	2 hosts	25 47 39	11	---
	weaveworksdemos/orders	0.4.7	master-demo2-tibuild-demo-twistlo...	23 40 38	11	---
	weaveworksdemos/carts	0.4.8	2 hosts	23 40 38	11	---
registry.infra.svc.cluster.local:5000	vulnerables/web-dvwa	latest	master-demo2-tibuild-demo-twistlo...	111 88 69 33	10	---
	splunk/splunk	7.1.2	master-demo2-tibuild-demo-twistlo...	51 33 46 33	11	---
	sonatype/nexus3	latest	master-demo2-tibuild-demo-twistlo...	9 25 22 31	11	---
registry.infra.svc.cluster.local:5000	infra/my_jenkins	latest	master-demo2-tibuild-demo-twistlo...	194 71 62 28	11	---
	jenkins/jenkins	2.150.3	master-demo2-tibuild-demo-twistlo...	187 71 62 28	10	---
	tl_demo/struts2_demo	2.3.12_build	master-demo2-tibuild-demo-twistlo...	104 8 29 16	11	---
	postgres	10.6	2 hosts	92 26 22 14	10	---
	weaveworksdemos/catalogue-db	0.3.0	2 hosts	63 266 45 13	11	---
	weaveworksdemos/user-db	0.4.0	master-demo2-tibuild-demo-twistlo...	45 32 33 13	11	---
	rabbitmq	3.6.8	2 hosts	56 33 28 13	11	---
	httpd	latest	master-demo2-tibuild-demo-twistlo...	54 106 49 12	9	---
registry.infra.svc.cluster.local:5000	infra/portal_httpd	latest	2 hosts	54 106 49 12	9	---
	openfrontier/openldap-server	latest	2 hosts	52 9 13 12	10	---

- The goal of the *Vulnerabilities* legend is to provide an extensive overview of what **Common Vulnerabilities and Exposures** exist, the **severity (low, medium, or high)**, **Common Vulnerability Scoring System (CVSS)**, **package, version, status**, when it was **published, discovered** and a **description** of what was discovered about the vulnerability.

CVE	SEVERITY	CVSS	PACKAGE	VERSION	STATUS	PUBLISHED	DISCOVERED	DESCRIPTION
CVE-2020-10531	high	8.80	icu	63.1-6	open	8 days	< 1 hour	An issue was discovered in International Components for Unicode (ICU) for C/C++ through 66.1. An integer overflow, leading to a heap-based buffer over...
CVE-2018-12886	high	8.10	gcc-8	8.3.0-6	open	> 10 months	< 1 hour	stack_protect_prologue in cfgexpand.c and stack_protect_epilogue in function.c in GNU Compiler Collection (GCC) 4.1 through 8 (under certain circumsta...
CVE-2019-20367	low	9.10	libbsd	0.9.1-2	open	72 days	< 1 hour	nlist.c in libbsd before 0.10.0 has an out-of-bounds read during a comparison for a symbol name from the string table (strtab).
CVE-2017-17942	low	8.80	tiff	4.1.0+git191117-2~deb10u1	open	> 2 years	< 1 hour	In LibTIFF 4.0.9, there is a heap-based buffer over-read in the function PackBitsEncode in tif_packbits.c.
CVE-2019-17543	low	8.10	lz4	1.8.3-1	open	> 5 months	< 1 hour	LZ4 before 1.9.2 has a heap-based buffer overflow in LZ4_write32 (related to LZ4_compress_destSize), affecting applications that call LZ4_compress_fas...
CVE-2019-13627	low	8.10	libcrypt20	1.8.4-5	open	> 5 months	< 1 hour	It was discovered that there was a ECDSA timing attack in the libcrypt20 cryptographic library. Version affected: 1.8.4-5, 1.7.6-2+deb9u3, and 1.6.3-...
CVE-2017-6363	low	8.10	libgd2	2.2.5-5.2	open	22 days	< 1 hour	** DISPUTED ** In the GD Graphics Library (aka LibGD) through 2.2.5, there is a heap-based buffer over-read in tiffWriter in gd_tiff.c. NOTE: the vend...

- The lab is now complete; you may end your reservation.