CLOUD SECURITY FUNDAMENTALS V2

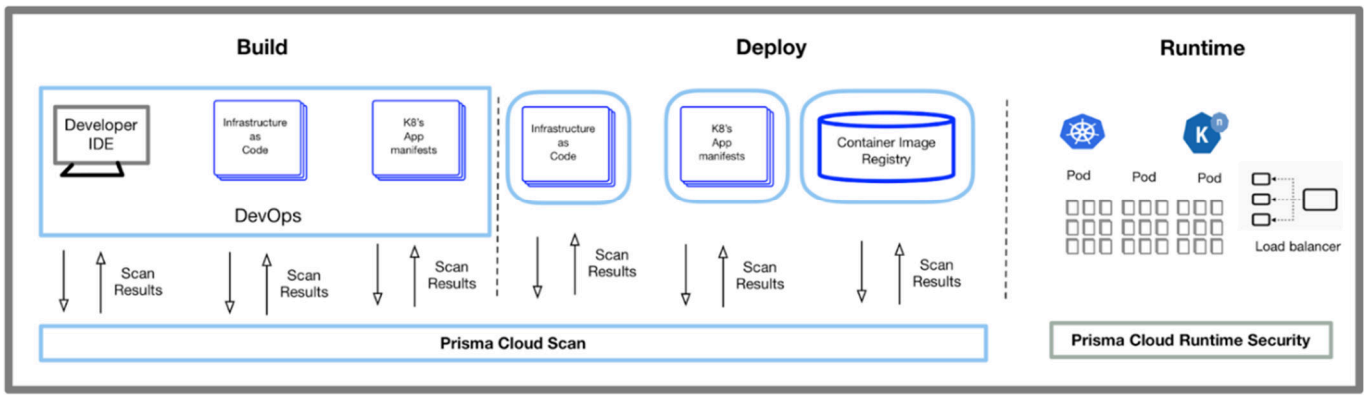
Lab 6: Container Vulnerability Scanning

Document Version: 2022-12-22

Contents

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.

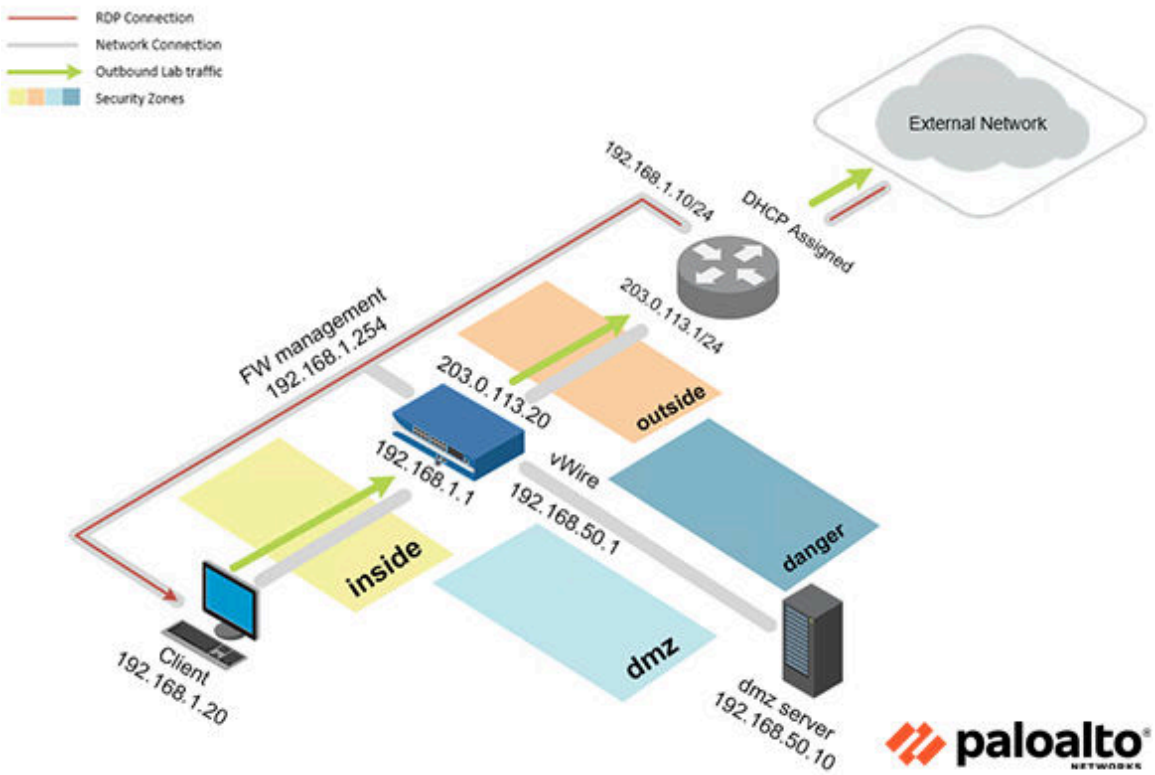


Objective

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.

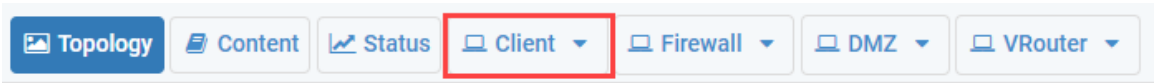
Virtual Machine	IP Address	Account (if needed)	Password (if needed)
Client	192.168.1.20	`lab-user`	`Pal0Alt0!`
DMZ	192.168.50.10	`root`	`Pal0Alt0!`
Firewall	192.168.1.254	`admin`	`Pal0Alt0!`

1 Container Vulnerability Scanning

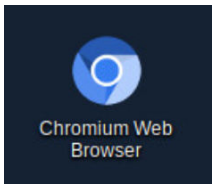
1.0 Load Lab Configuration

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

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

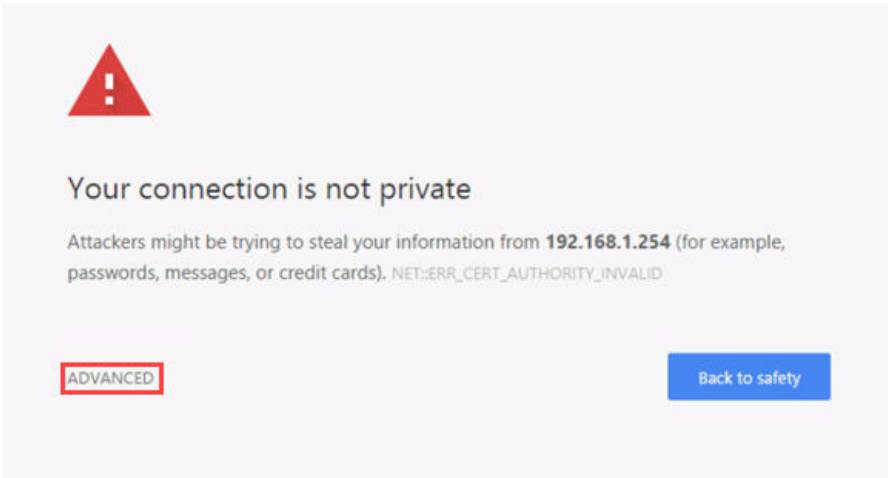


2. Log in to the Client PC as username `lab-user`, password `Pal0Alt0!`.
3. Double-click the **Chromium Web Browser** icon located on the Desktop.



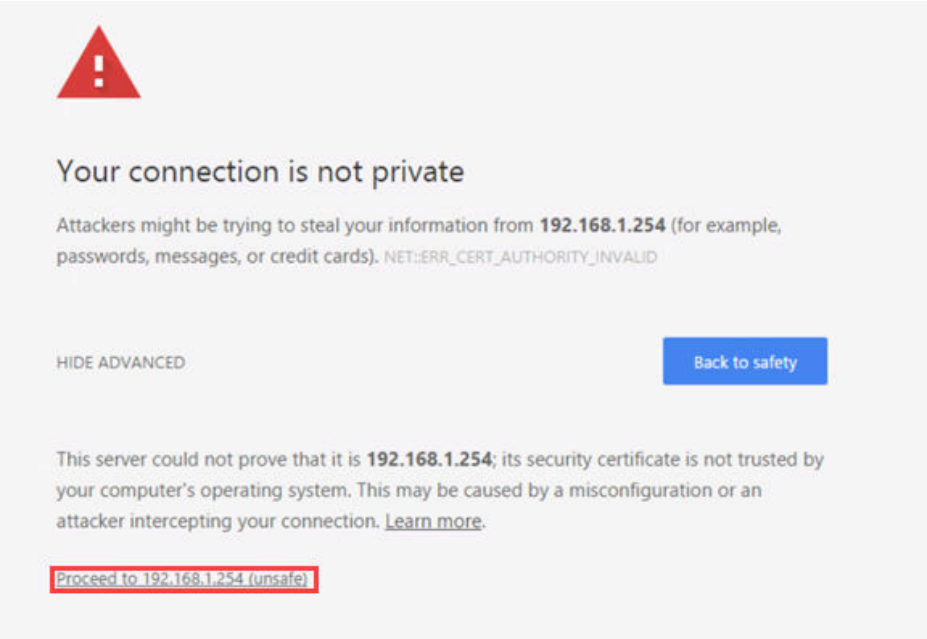
4. In the *Chromium* address field, type `https://192.168.1.254` and press **Enter**.

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



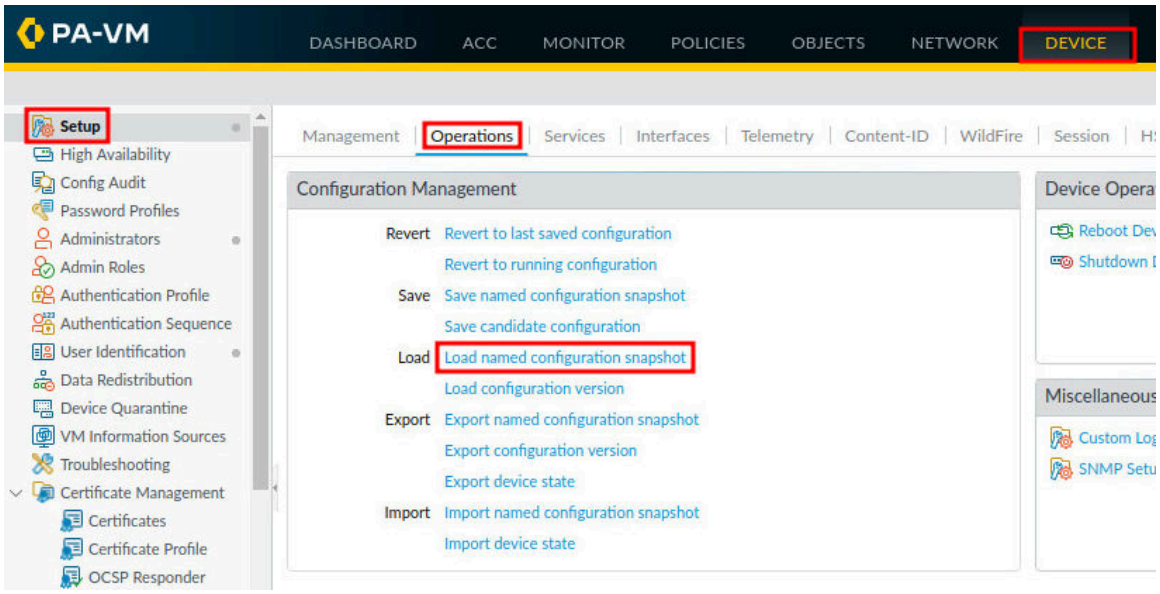
If you experience the "Unable to connect" or "502 Bad Gateway" message while attempting to connect to the specified IP above, please wait an additional 1-3 minutes for the Firewall to fully initialize. Refresh the page to continue.

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

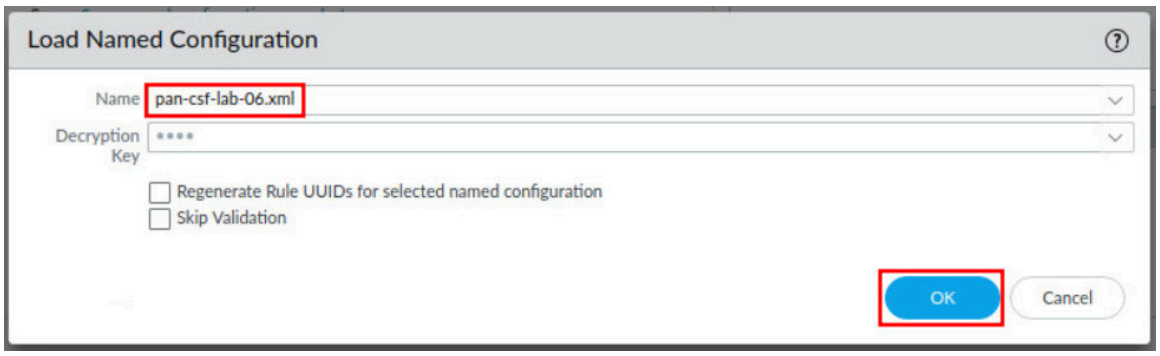


7. Log in to the Firewall web interface as username `admin`, password `Pal0Alt0!`.

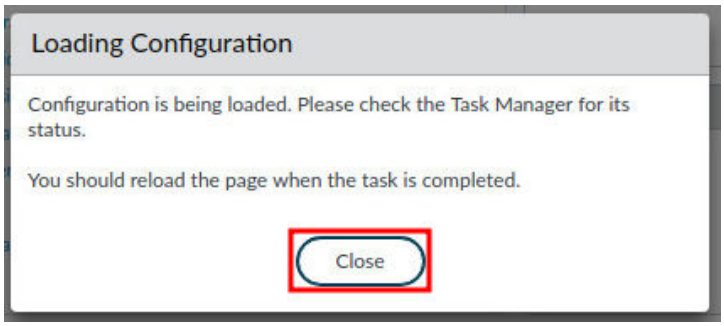
8. In the web interface, navigate to **Device > Setup > Operations** and click on **Load named configuration snapshot** underneath the *Configuration Management* section.



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

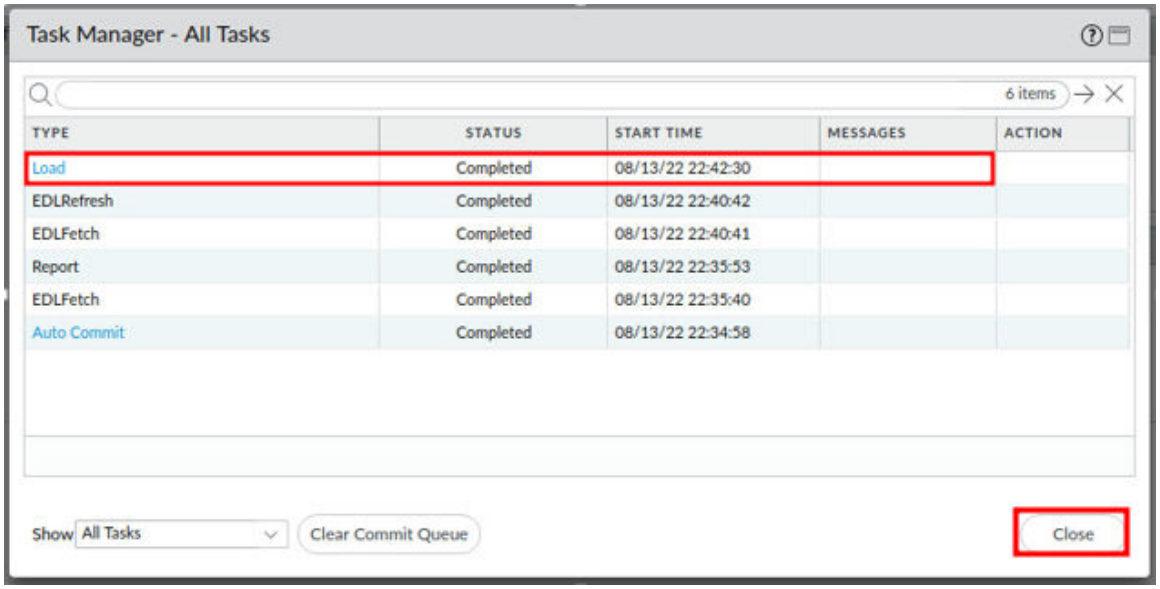


10. In the Loading Configuration window, a message will show *Configuration is being loaded. Please check the Task Manager for its status. You should reload the page when the task is completed.* Click **Close** to continue.

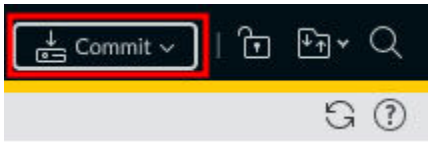


11. Click the **Tasks** icon located at the bottom-right of the web interface.

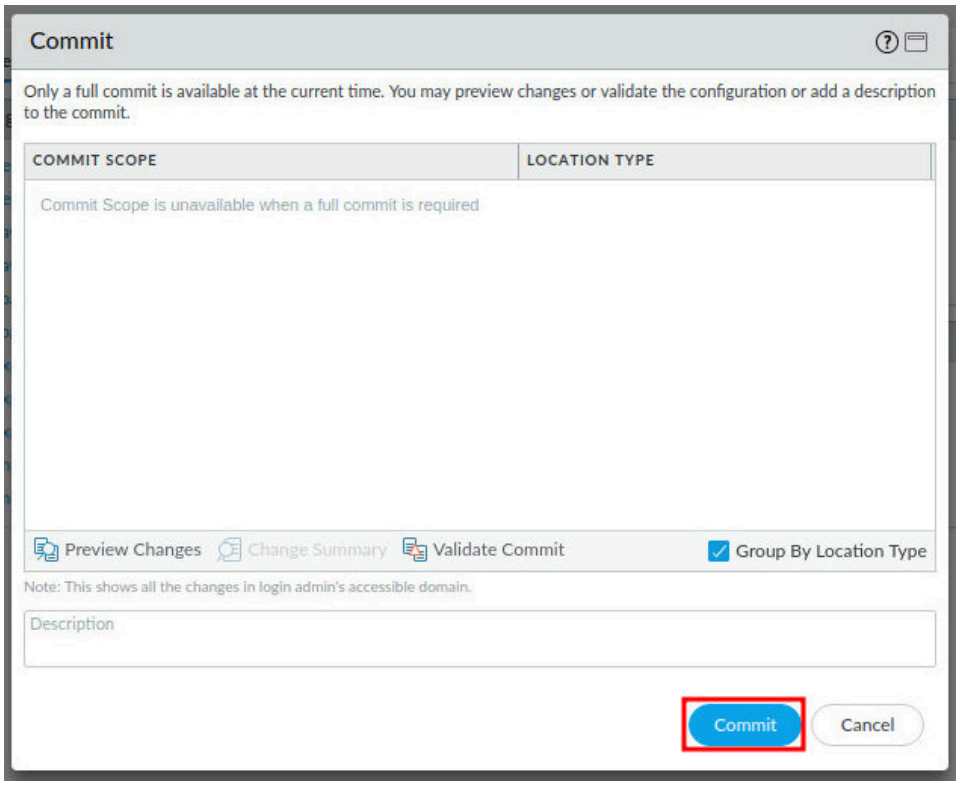
12. In the *Task Manager – All Tasks* window, verify the *Load* type has successfully completed. Click **Close.**



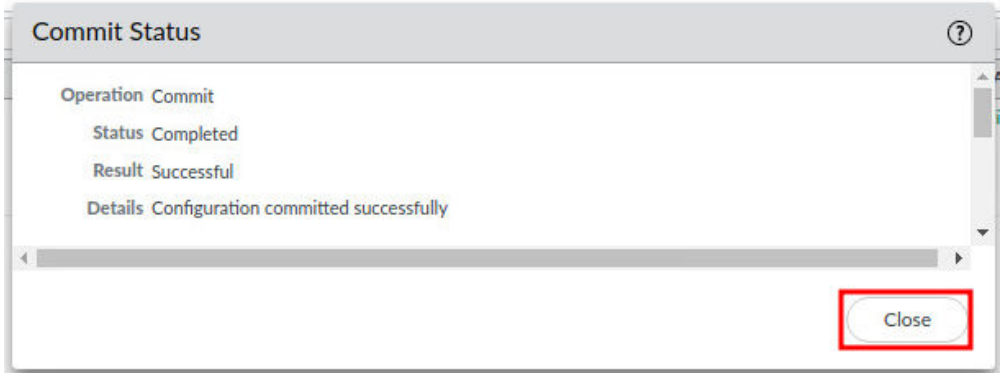
13. Click the **Commit** link located at the top-right of the web interface.



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.

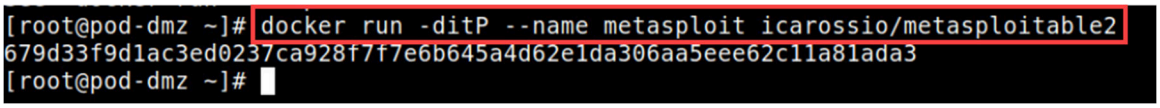
1. Launch **Xfce** Terminal in the lower-left of the student *Desktop*.

2. SSH to the *DMZ* by typing the command below. Use `Pal0Alt0!` 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
```



This output will display all the DMZ server ports mapped to the metaspoitable 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.

6. 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                                                                    csf-dvwa
679d33f9d1ac        icarossio/metasploitable2      "/bin/sh -c '/bin/se…"   7 minutes ago        Up 7 minute
s        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/tc
p, 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

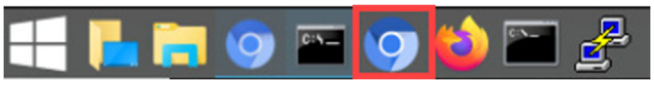
As you can see from the output of the command, container "**csf-dvwa**" using image vulnerables/web-dvwa with the DMZ server host port 8080 mapped to the container port 80 has been up for about a minute. Your results may vary on the uptime.

7. Close the SSH connection to the DMZ server by typing `exit`.

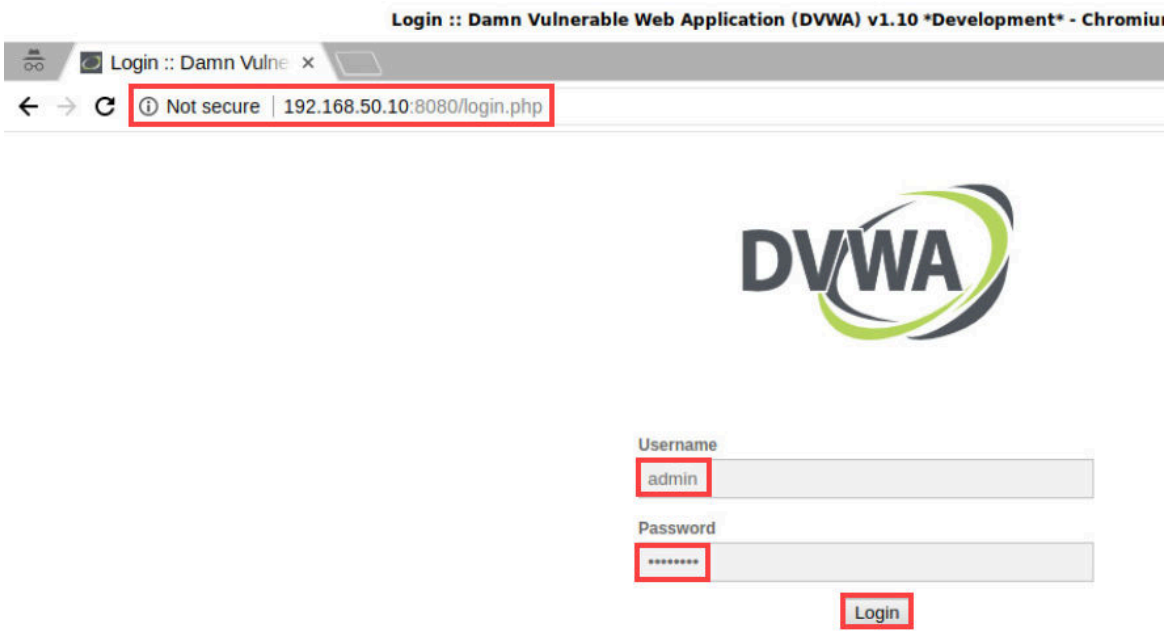
```
[root@pod-dmz ~]# exit
```

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

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

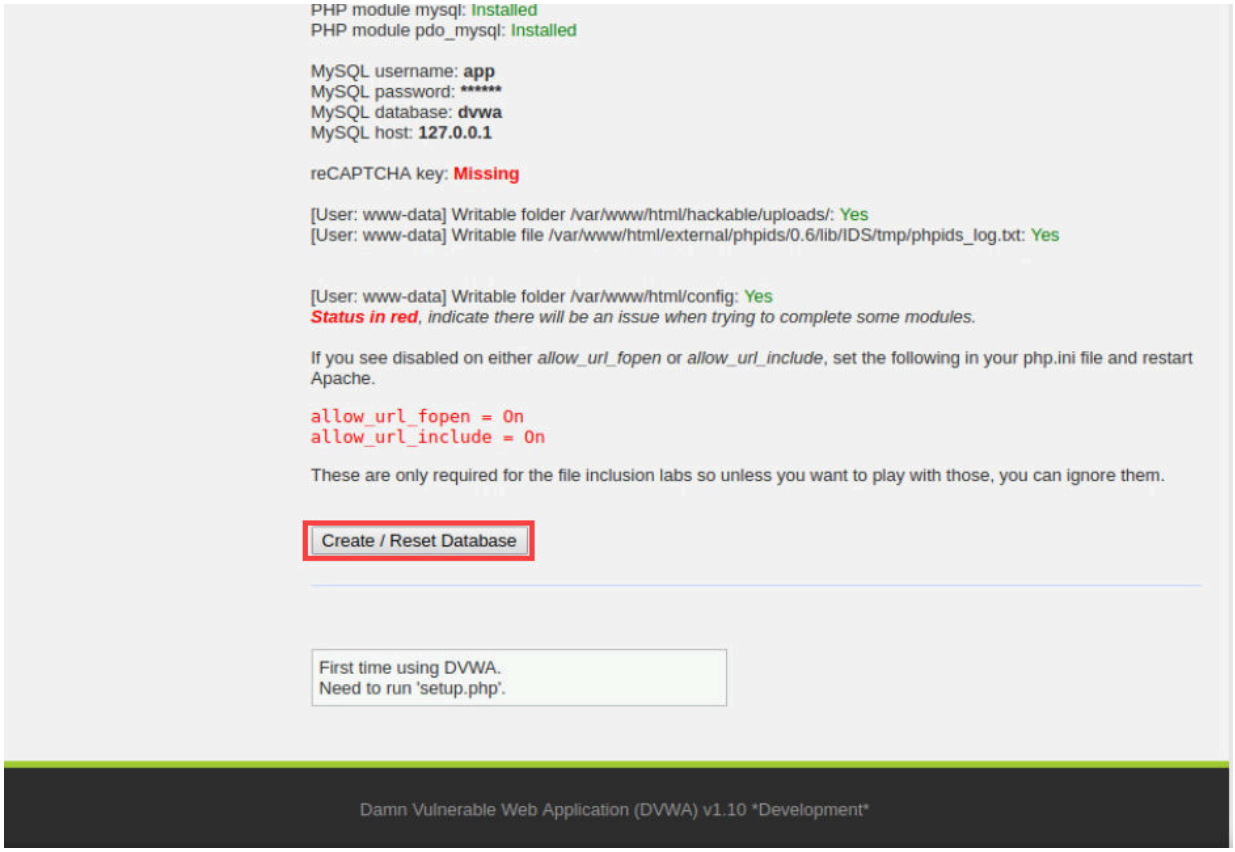


9. Access the *DVWA* website by entering the URL `http://192.168.50.10:8080` and press **Enter**. On the DVWA webpage, enter `admin` for the *username* and **password** for the *password*. Click **Login**.

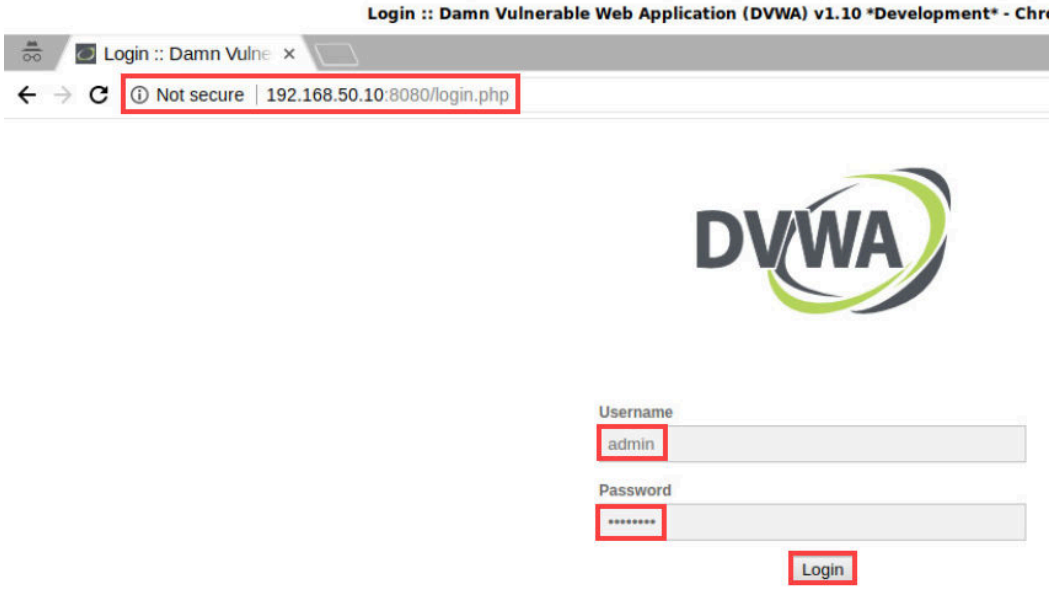


DVWA is a platform that can be used to learn how to conduct sql injections, cross site scripting and brute force attacks. There are multiple websites on the internet that will show you how to conduct web attacks against DVWA, here is one resource: http://www.dvwa.co.uk/ . For this lab, you're going to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

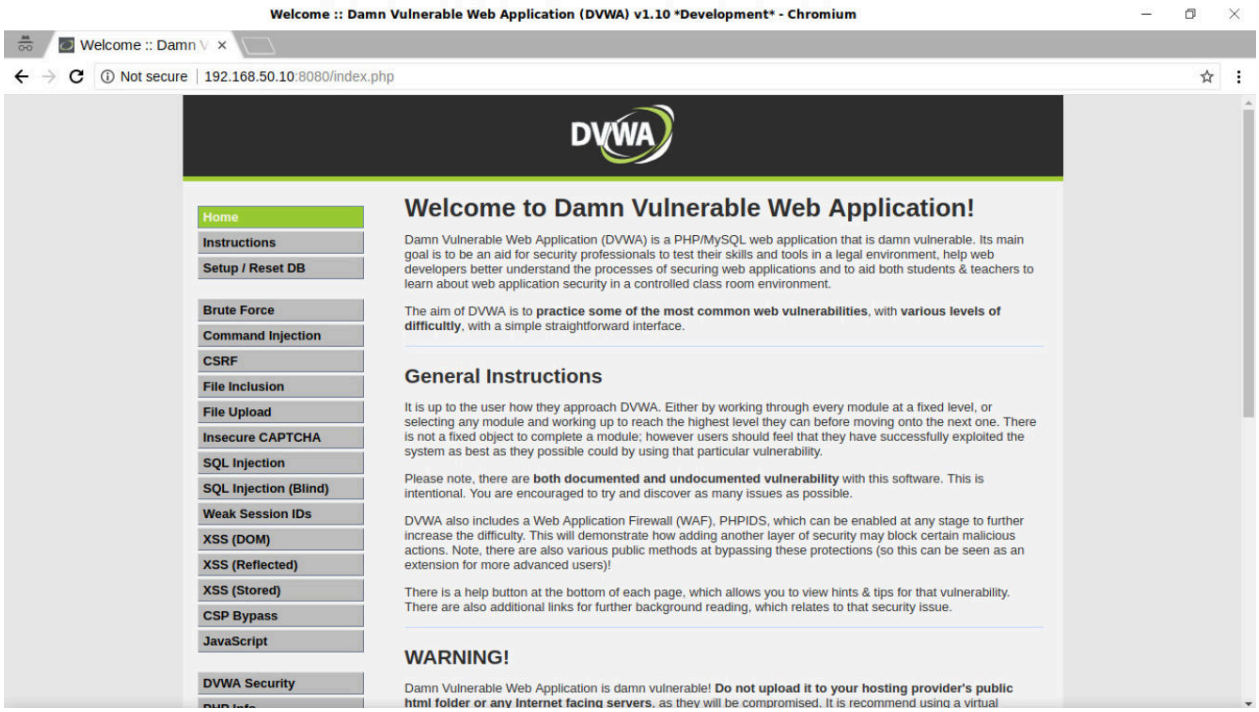
10. On the *DVWA Database Setup* page, scroll to the bottom of the page and click the **Create / Reset Database** button.



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

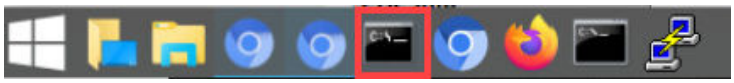


12. On the *DVWA* webpage, review the website after creating the database. Do not change any settings.



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.

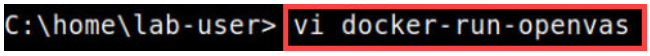


1.2 Use Docker Compose to Create an OpenVas Scanning Container

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.

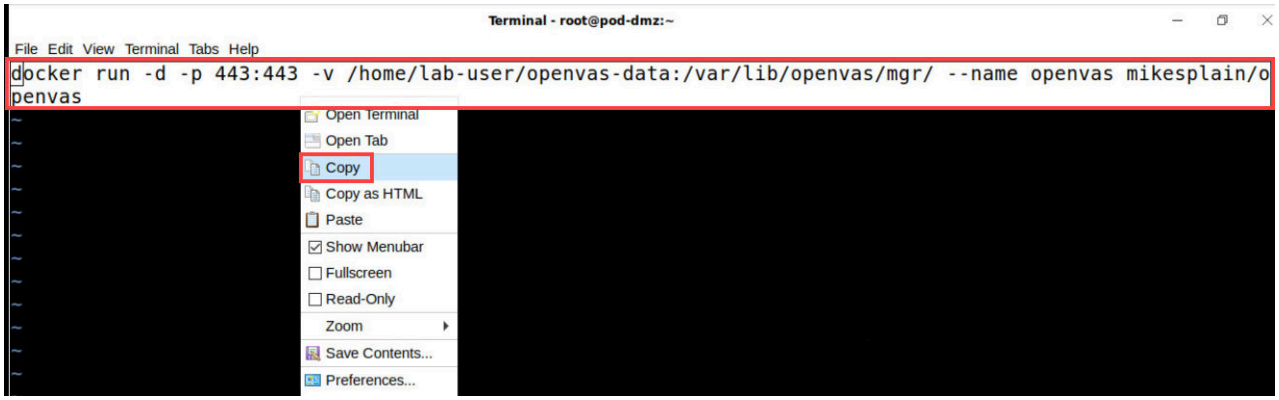
1. To view the docker run command that was used to create the docker-compose.yml, enter the command below.

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

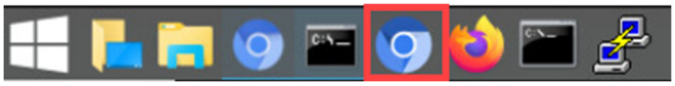


This docker run command creates a container named openvas using the docker image mikesplain/openvas. The command maps the client port 443 to the container port 443 and maps the client's /home/lab-user/openvas-data directory/volume to the container's /var/lib/openvas/mgr directory/volume. By mapping a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

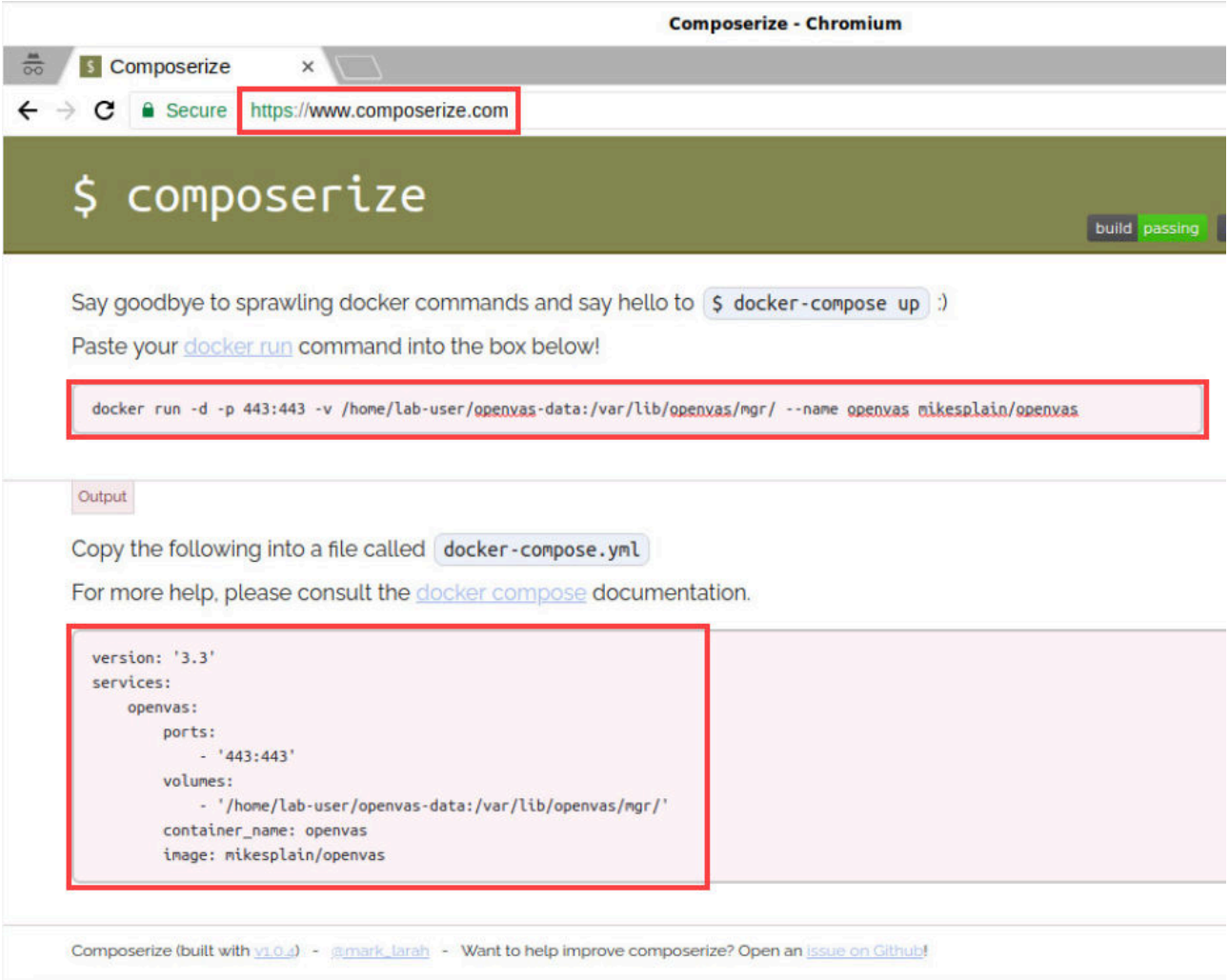
2. Highlight the **docker run** command from the docker-run-openvas file displayed on your terminal. After selecting the docker run command, right-click the run command and click **Copy**.



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

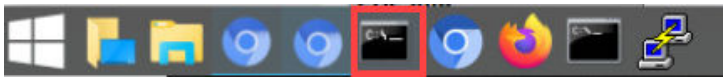


4. Enter the following URL `https://www.composerize.com` and click **Enter**. Paste the Docker run command from **step 2** into the website's docker run text box and view the output of the yaml file.

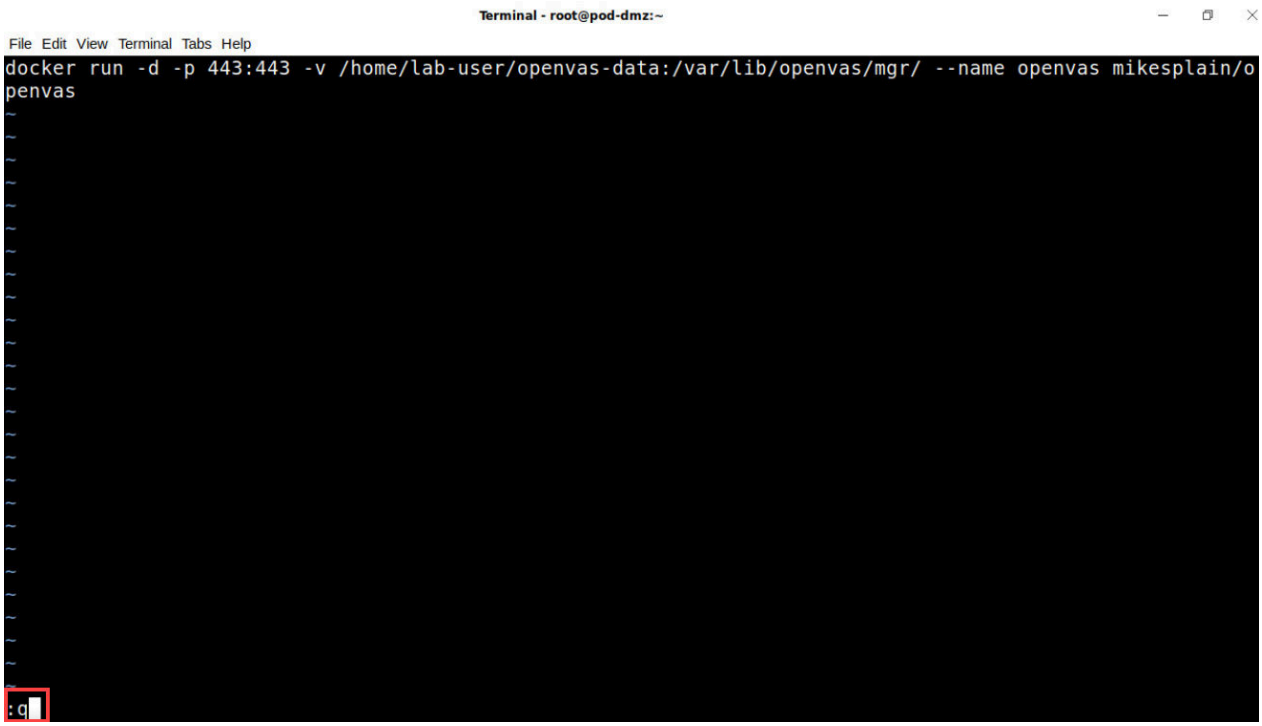


5. Close the *Composerize – Chromium* window by clicking the **X** icon located at the top-right.

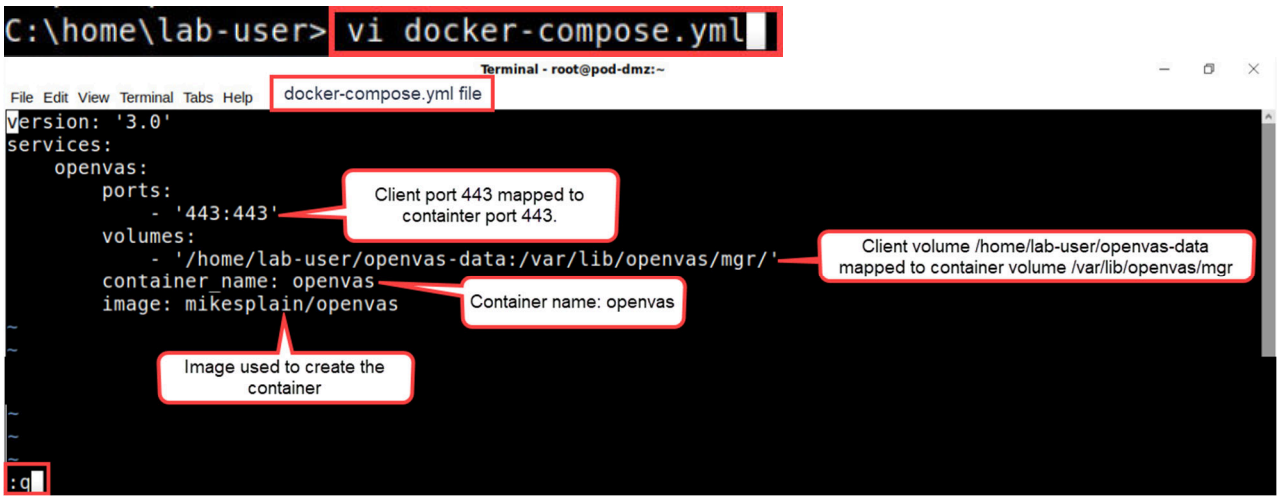
6. Change focus back to the *terminal window* by clicking on the **Terminal** icon located in the lower-left of the student *Desktop*.



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

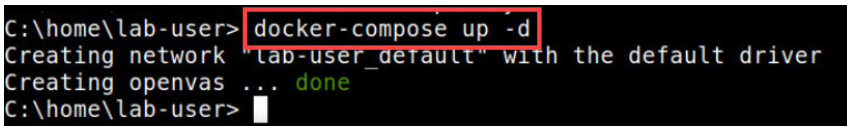


8. To view the contents of the *docker-compose-yml* file, enter the command below. After viewing the contents of the *docker-compose-yml* file, type `:q` and press **Enter**.



9. Run the OpenVas container by typing the command below.

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



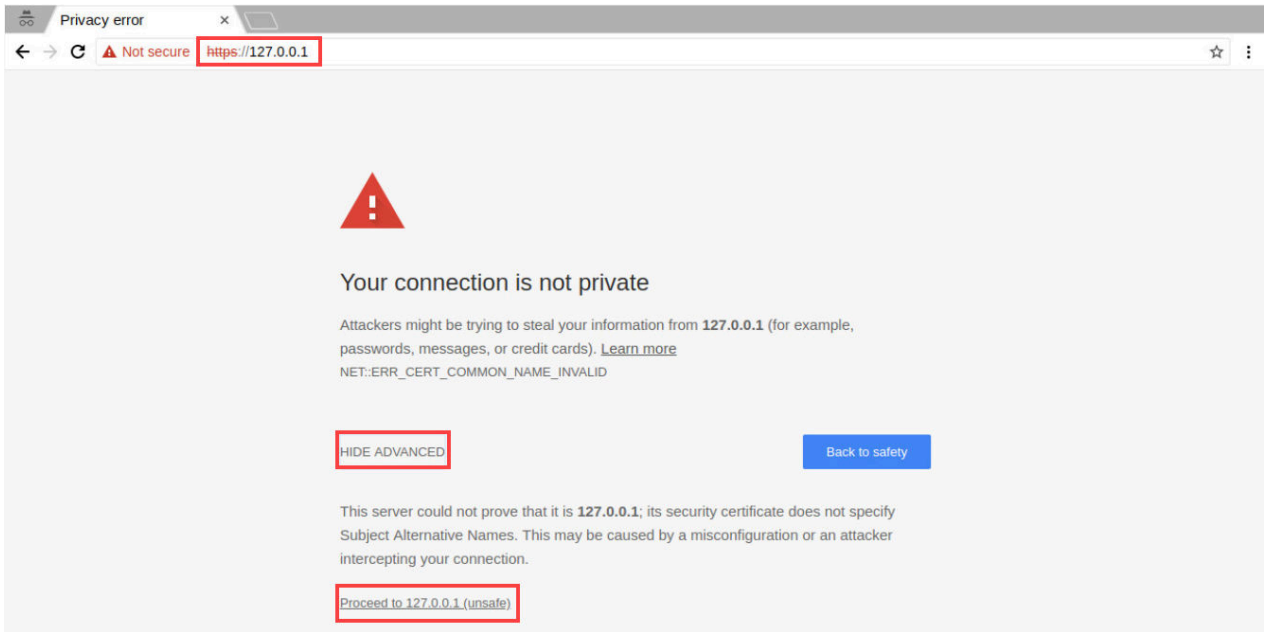
10. Open a new *Chromium browser* by clicking on the **Chromium** icon located in the lower-left of the student *Desktop*.

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

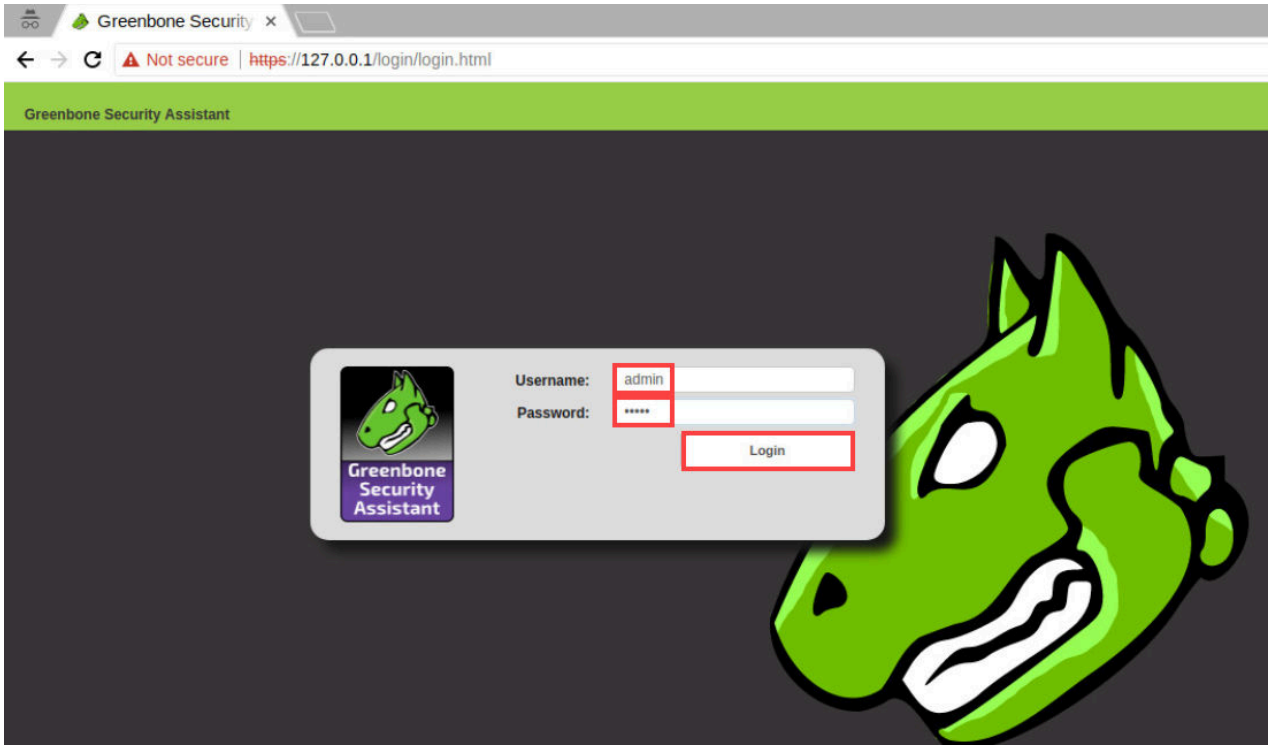
1.3 Conduct a Vulnerability Scan of the DMZ Server using an OpenVas Container

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

1. 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)**.

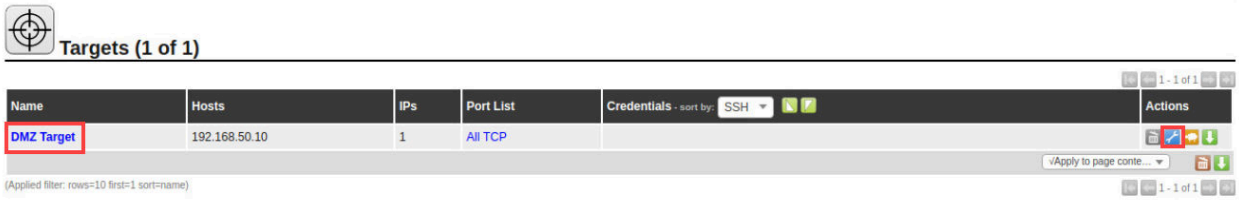


2. On the *OpenVas* website, enter `admin` for the *username* and `admin` for the *password*. Click **Login**.

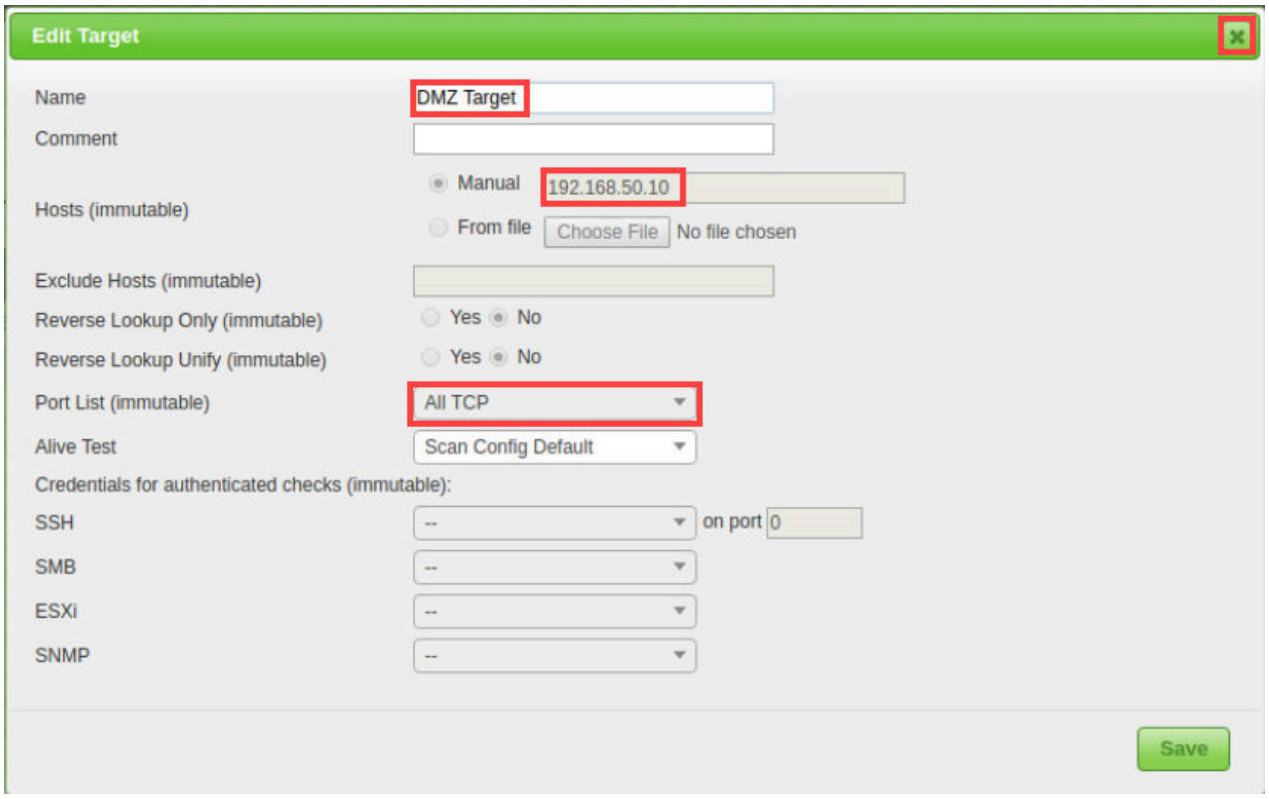


3. On the website's toolbar, click **Configuration > Targets** to access the configured targets.

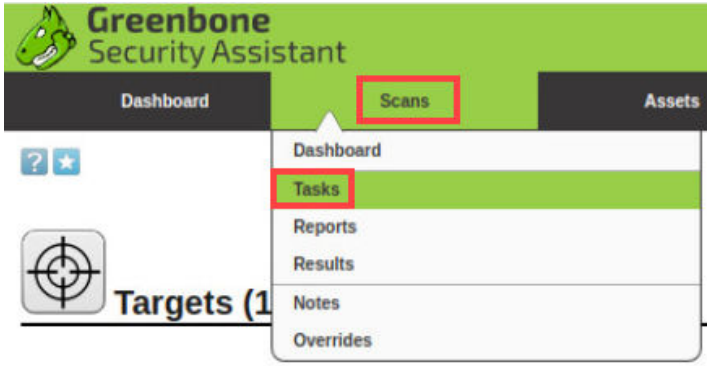
4. In the *Targets* webpage, click the **wrench** icon on the far-right of the **DMZ Target** to view the *configuration settings*.



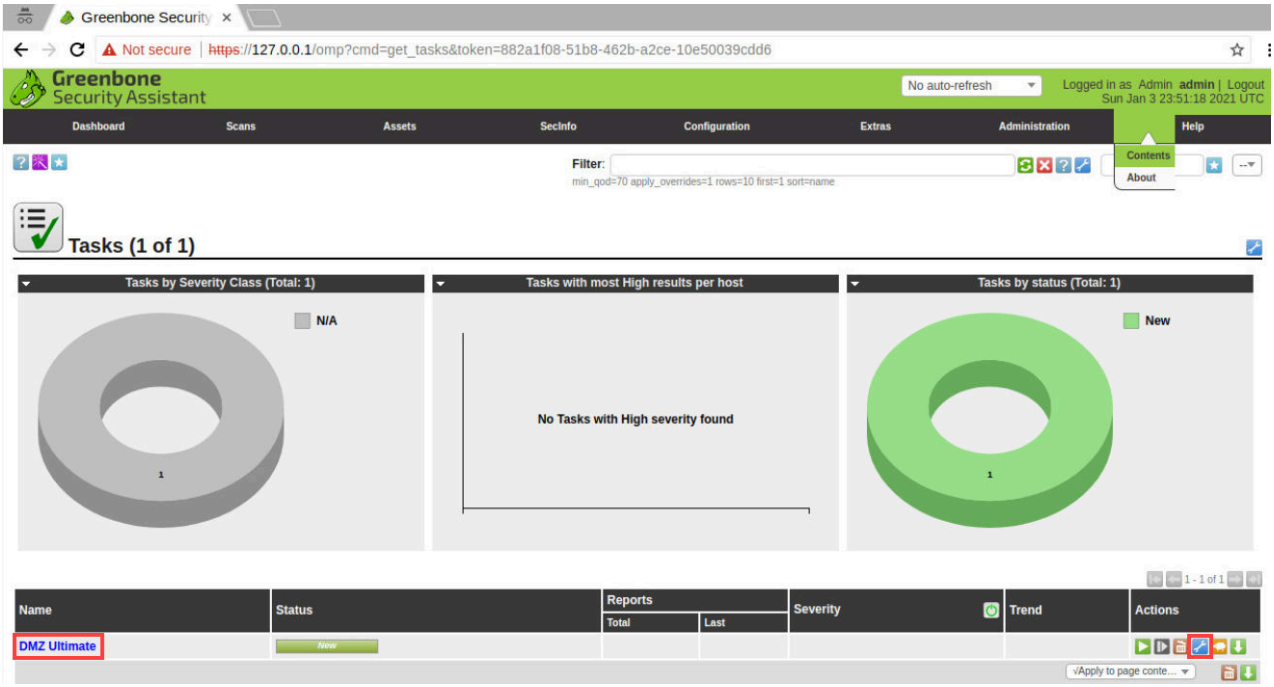
5. In the *Edit Target* dialogue box, observe the name **DMZ Target** and the *IP Address* of **192.168.50.10**. Lastly, notice that you are scanning all the **65,535 TCP ports**. After viewing the *Edit Target* details, click the **X** icon to close the window.



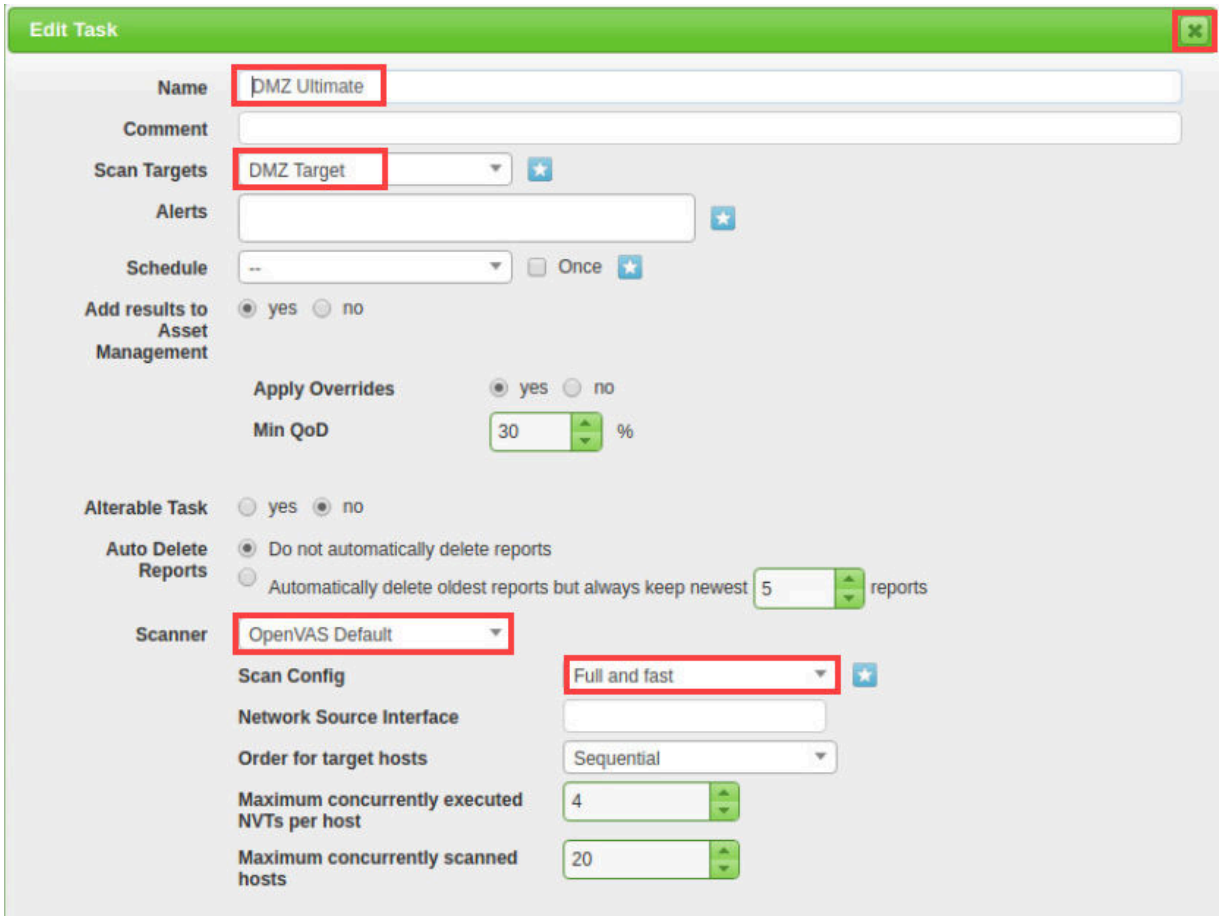
6. On the *website's* toolbar, click **Scans > Tasks** to access the *Tasks*.



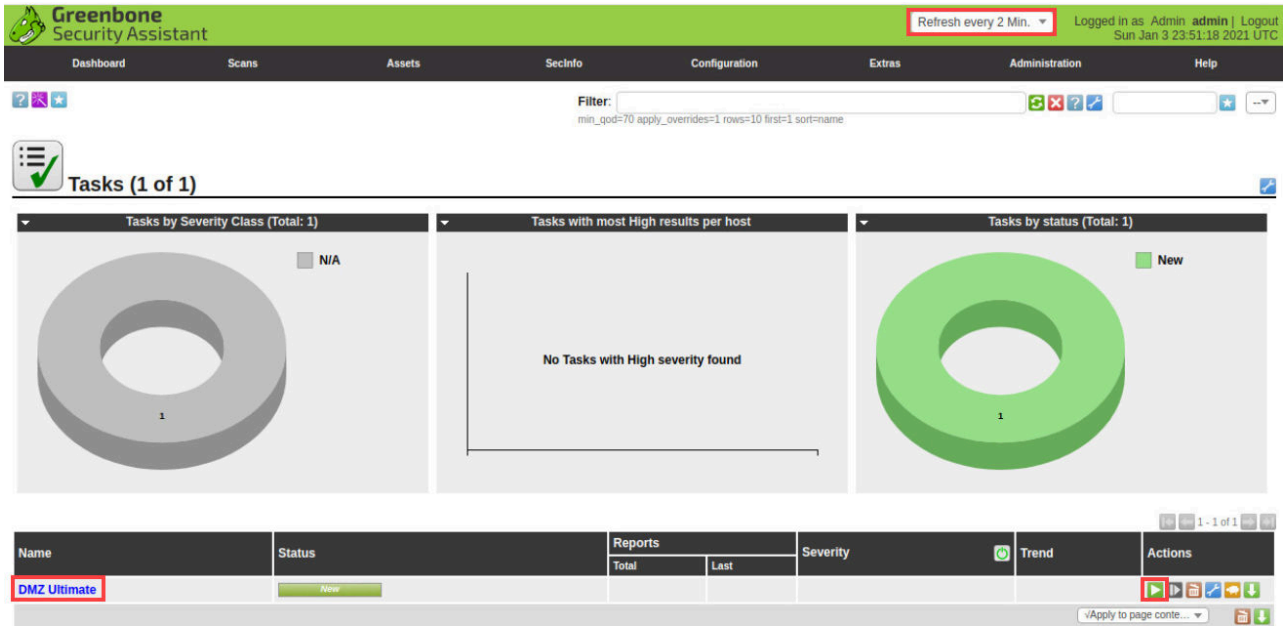
7. Scroll down to the bottom of the *task* window, click on the **wrench** icon for *DMZ Ultimate* and view the *configuration* settings.



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.



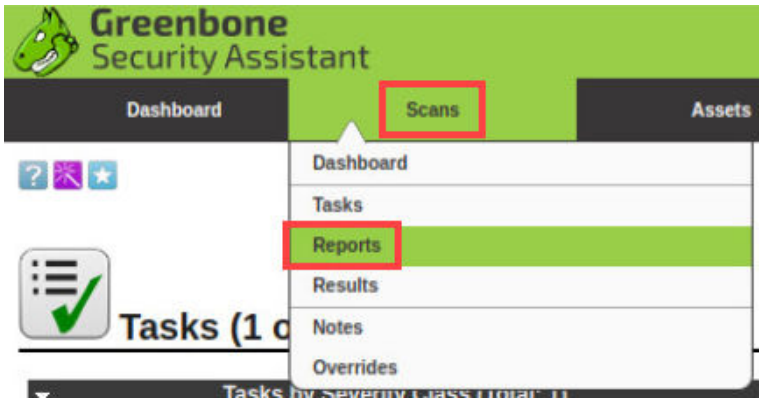
9. On the **Scans > Tasks** webpage, select **Refresh every 2 min** at the top-right. Lastly, select the **Play** icon to the far-right of **DMZ Ultimate** to start a scan.



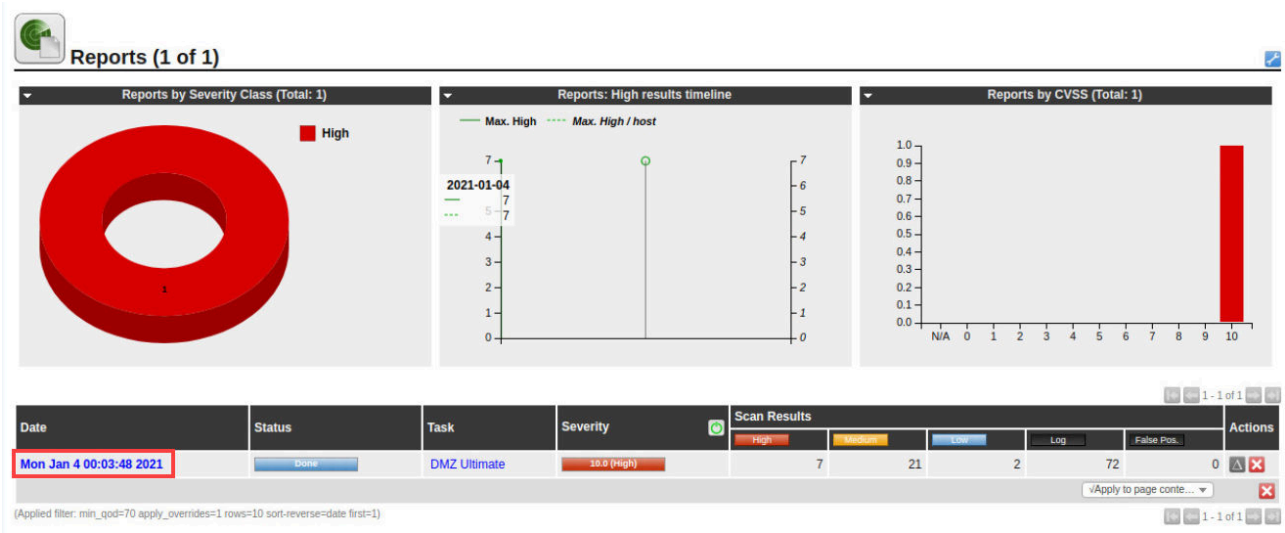
Please Note

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.

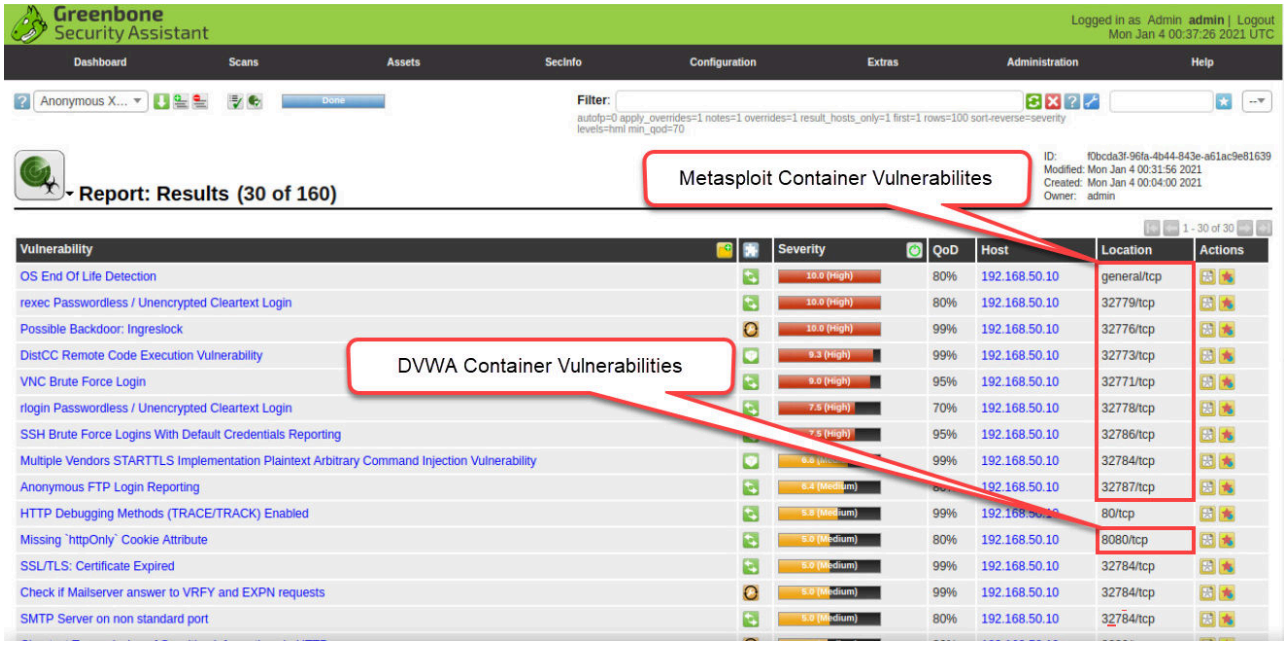
10. After the scan is complete, select **Scans > Reports**.



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.



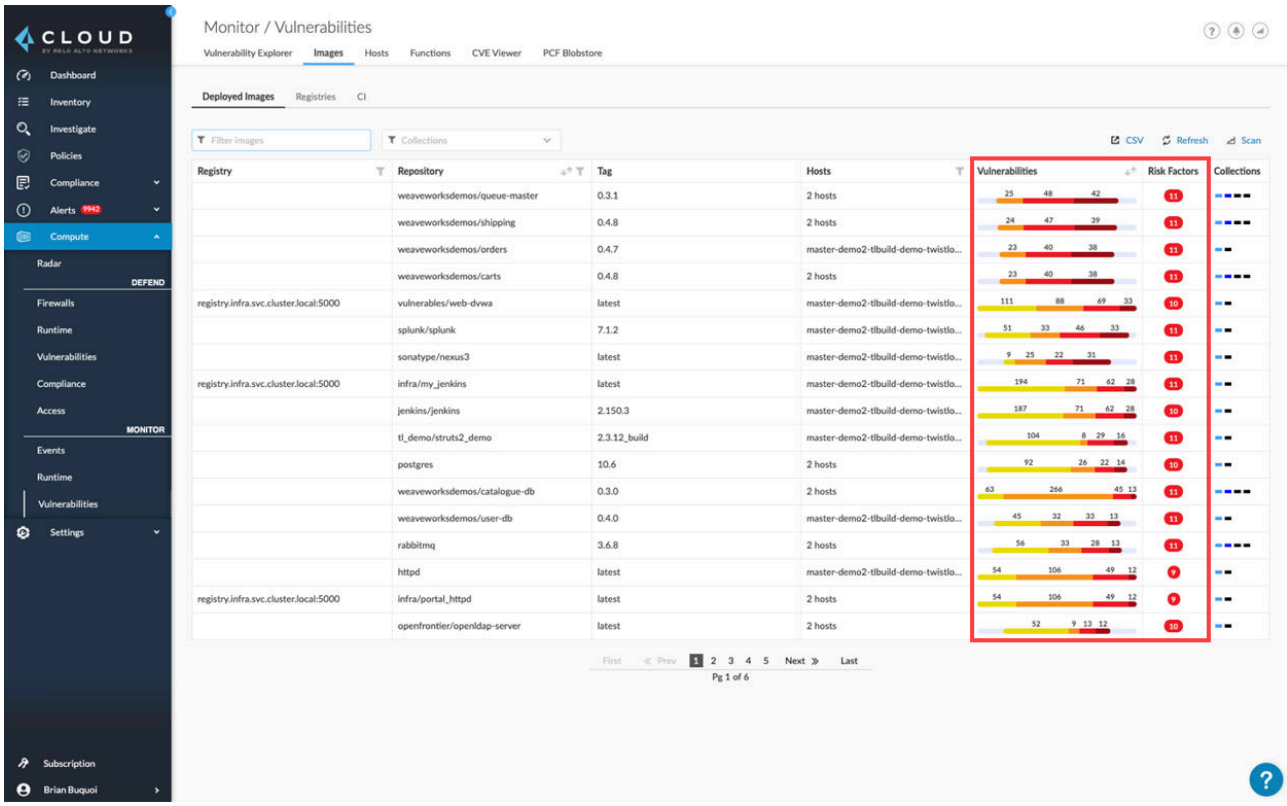
Please Note

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.



3. 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.

Vulnerabilities

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	libgcrypt20	1.8.4-5	open	> 5 months	< 1 hour	It was discovered that there was a ECDSA timing attack in the libgcrypt20 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...

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

