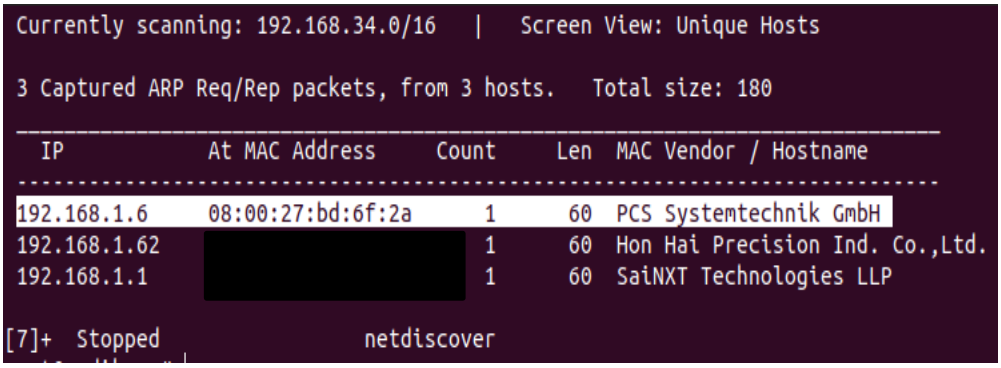
Vulnerability Assessment and Penetration Testing

CYBERSPLOIT-1

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

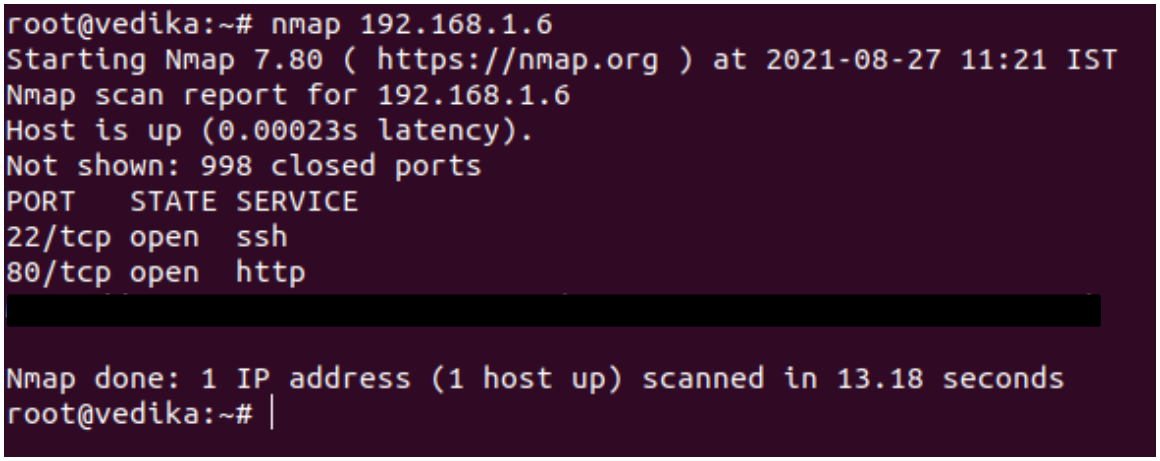
Submitted to: Wattlecorp Cybersecurity Labs.

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])



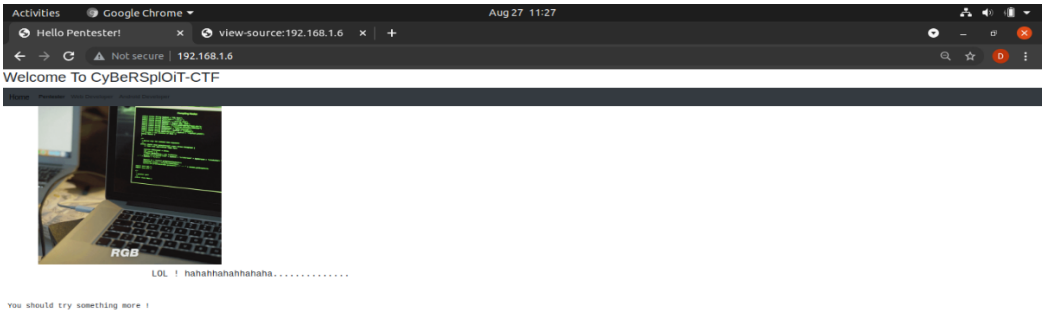
In the above screenshot, we have gotten a few IP addresses. It is ostensible, net discover can be stopped forcefully if we don't want a full scan; or otherwise. Here**, 192.168.1.6** is the required IP address.

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.



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.

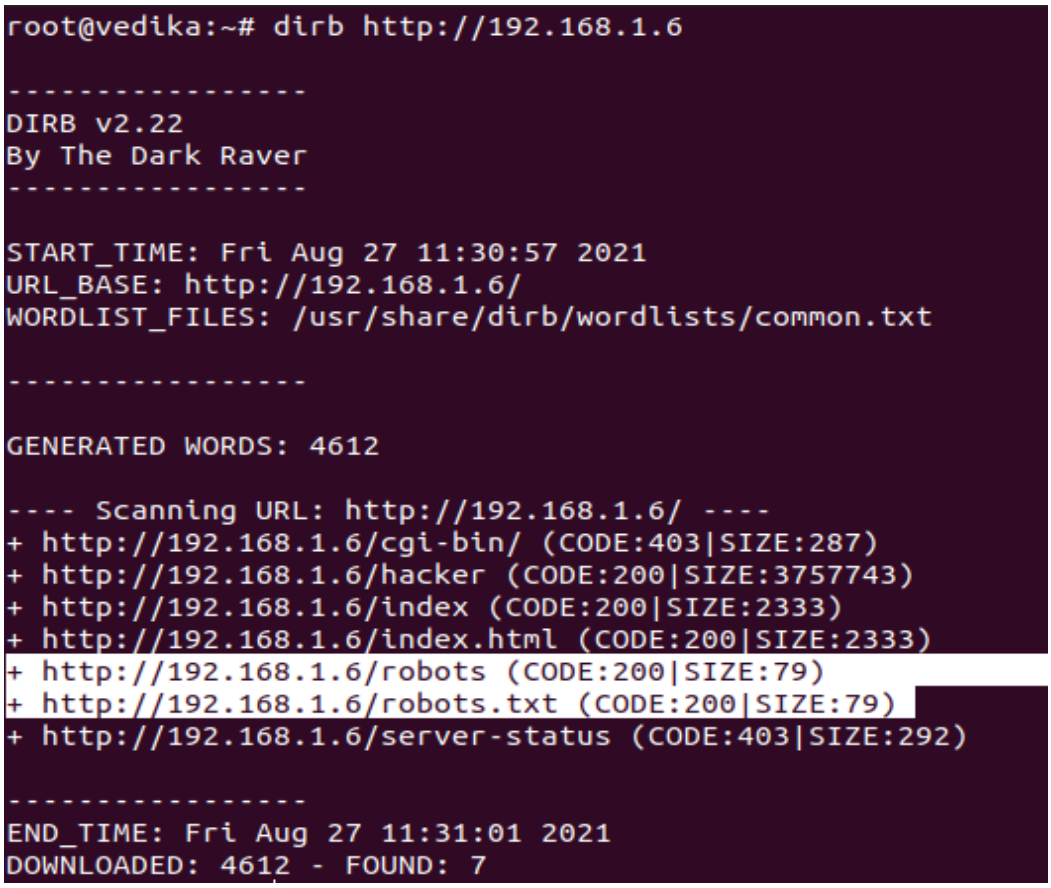
3. After looking it up on the browser, we get a clue. (Which is not really useful.) Since, “You should try something more”, it’s as plain as a day. **[Checking the page source code!]**



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/dis
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.
<pre>           <img src="hacker.gif" class="img-fluid" alt="hacke
</pre>
<pre>
<h4>                          LOL ! hahahhahahhahaha.....
                                                <h5> You shoulc
</pre>


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

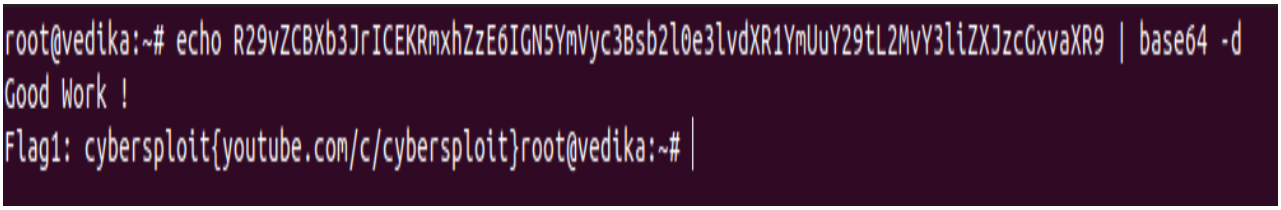
5. Very next step is to find a password. Here, we'll use **DIRB** command. DIRB is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary-based attack against a web server and analyzing the response. **Command: dirb <URL>**



DIRB's output gives us information about hidden web objects. And as we can see robots.txt and robots seems specious. 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**



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

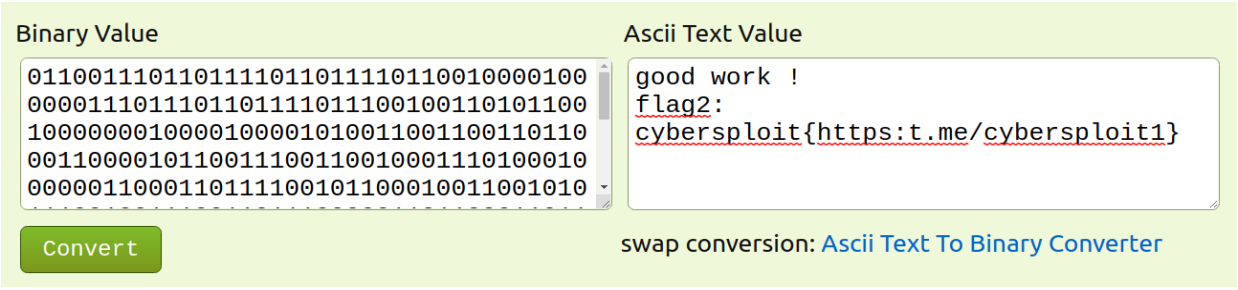
8. As port 22 is open, SSH service is available.
   Command: ssh itsskv@198.162.1.6
   After entering the above password, **we are in the targeted machine!**

```
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:~$
```

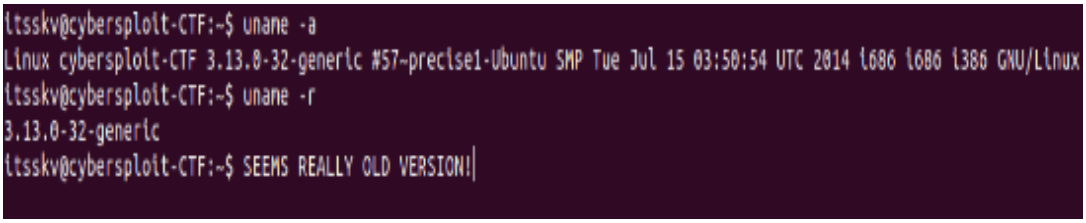
9. After performing some commands like **pwd, cd, ls** we get the flag no.2 as shown in following screenshot.

```
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 01110111 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 00111010 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**



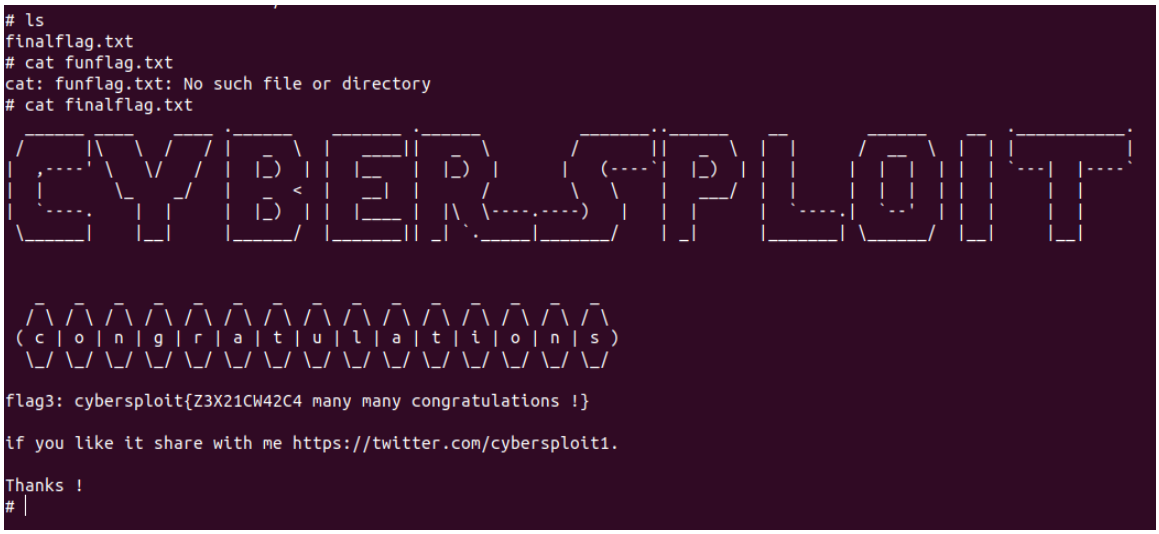
10. Well, very imminent step is to check **About** of the system. Using command **uname-a/uname-r,** we understand the version is older. It's astute that we might find a vulnerability regarding the same.



11. Upon researching about Linux kernel 3.13.0, found 'overlayfs' 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 'overlayfs' privileges escalation vulnerability allow local users to gain root privileges by taking advantage of configuration in which overlayfs 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!**



Stay safe.