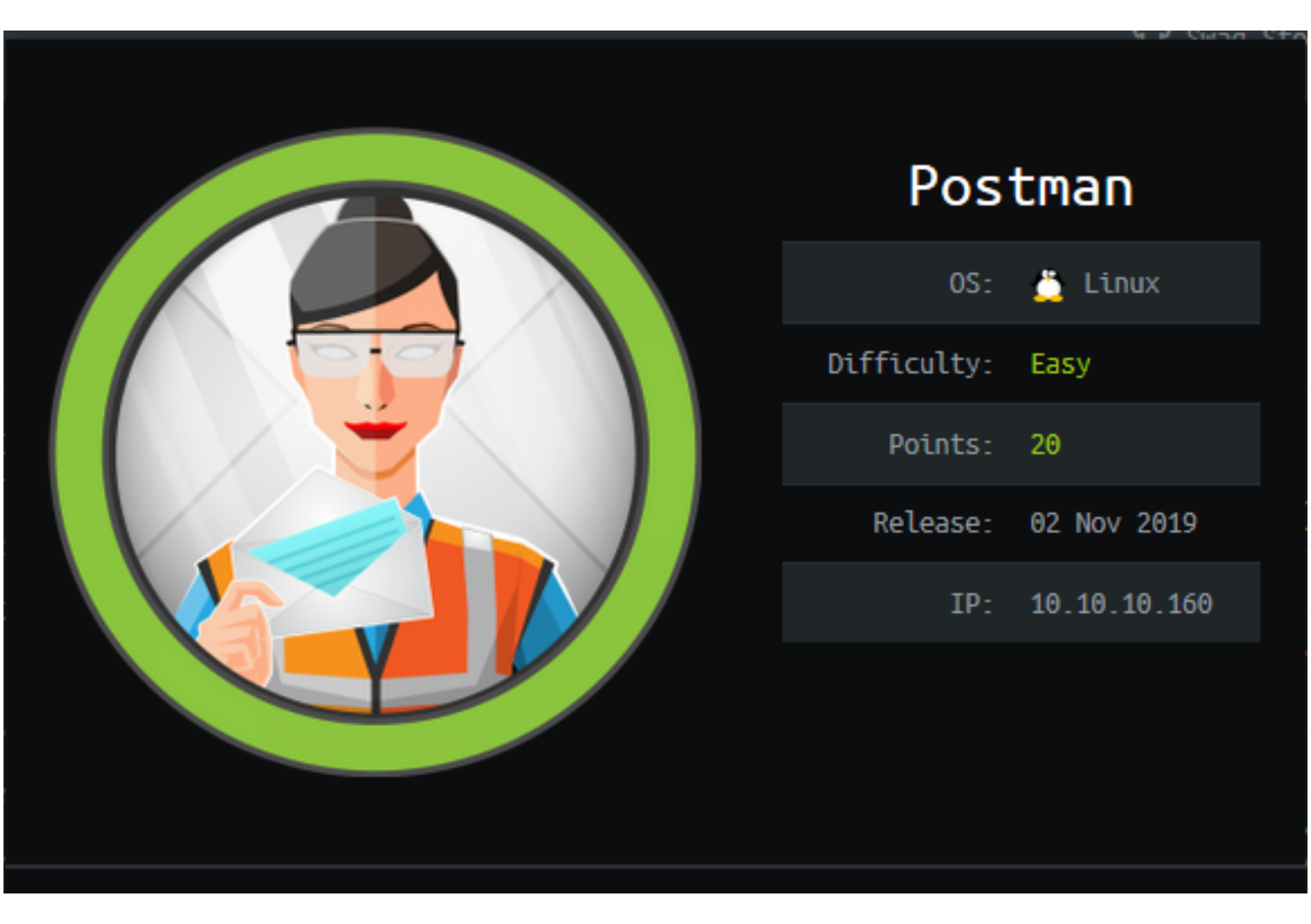
Postman



Nmap

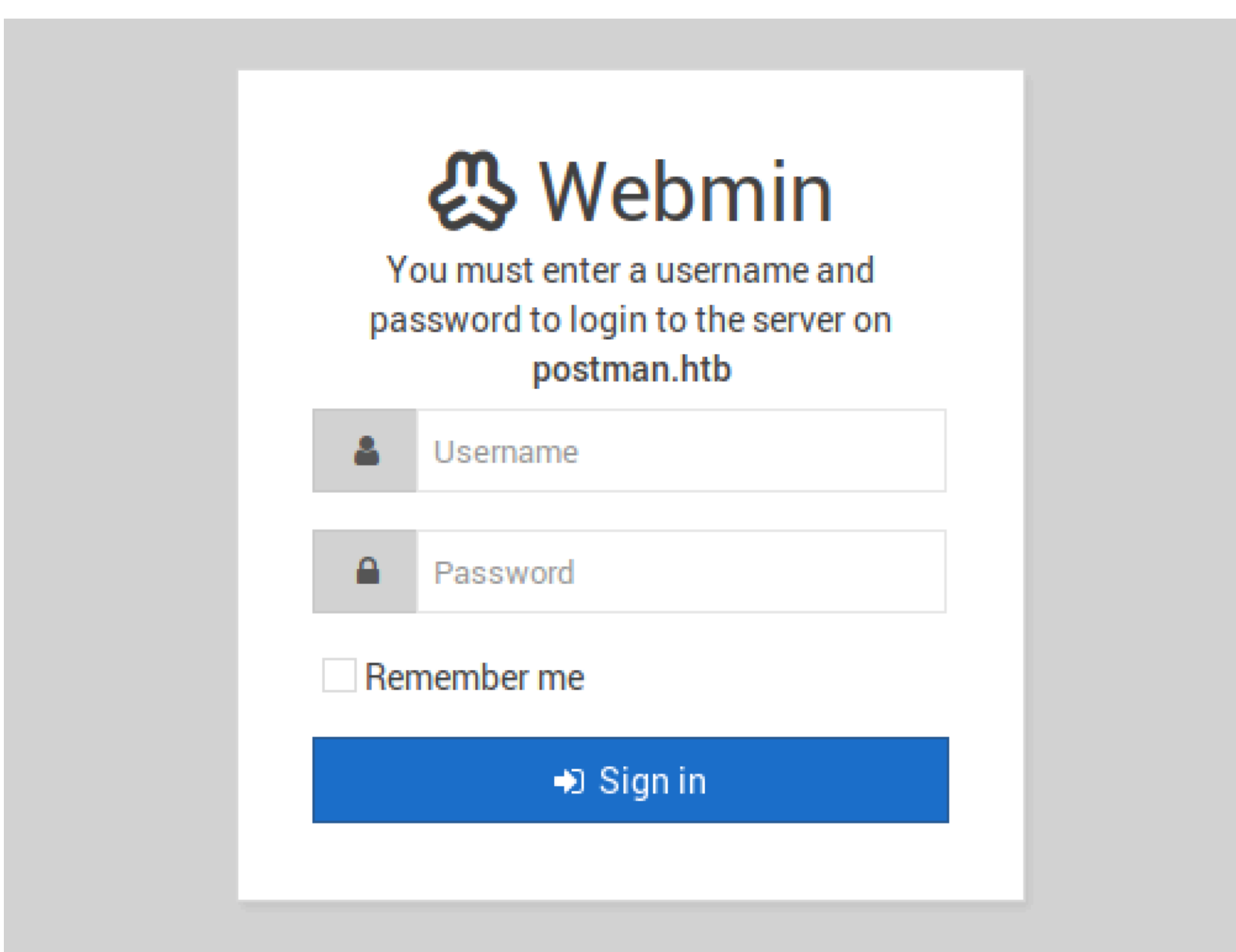
Ran a quick scan `nmap -T5 -Pn 10.10.10.16` 0, but found the number of ports suspiciously low and ran a deeper search: `nmap -T5 -Pn 10.10.10.160 -p- -A` , a good thing too as the first scan missed port `6379`

```
PORT      STATE SERVICE VERSION
22/tcp    open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3
 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http    Apache httpd 2.4.29 ((Ubuntu))
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: The Cyber Geek's Personal Website
6379/tcp  open  redis   Redis key-value store 4.0.9
10000/tcp open  http    MiniServ 1.910 (Webmin httpd)
|_http-title: Site doesn't have a title (text/html; Charset=iso-8859-1).
```

Whilst we enumerate the new ports, let's run Dirbuster in the background against port 80

Port 10000

Going to `10.10.10.160:10000` in our url leads to a suggested link that doesn't work, just change it to `https://10.10.10.60:10000` and it will work. Accept the certificate risk, and you'll be met with this page:



Let's try some **SQL injections** (via `burpsuite`), and let's search webmin **default creds**.

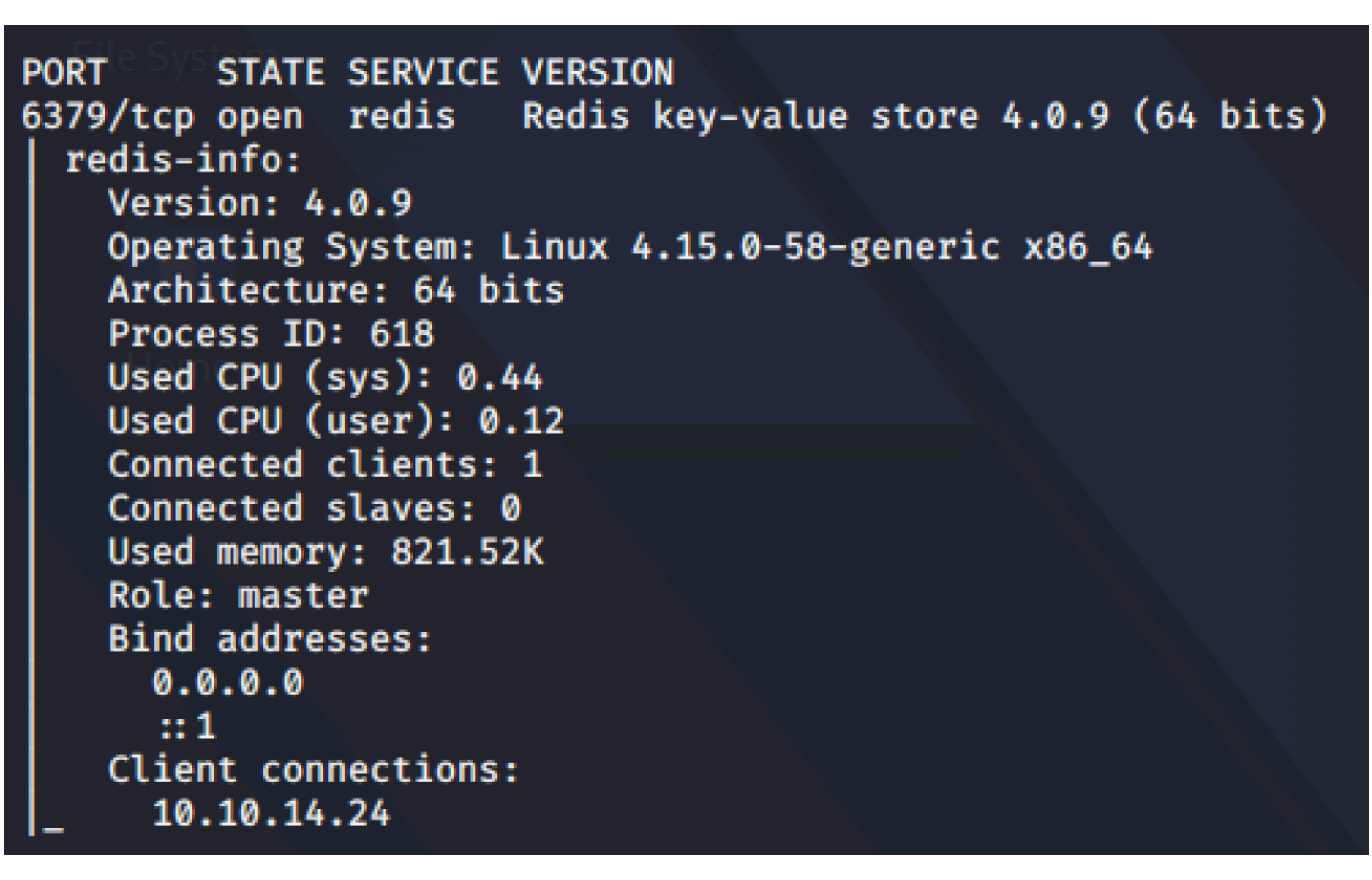
- **SQL Injection:** blocks our IP with too many failed authentication.; **default creds** didn't work. Let's enumerate some other ports.

New Port: Redis

https://book.hacktricks.xyz/pe

I've never seen port 6379, so I get some info on it: https://book.hacktricks.xyz/pentesting/6379-pentesting-redis

`nmap --script redis-info -sV -p 6379 10.10.10.160`

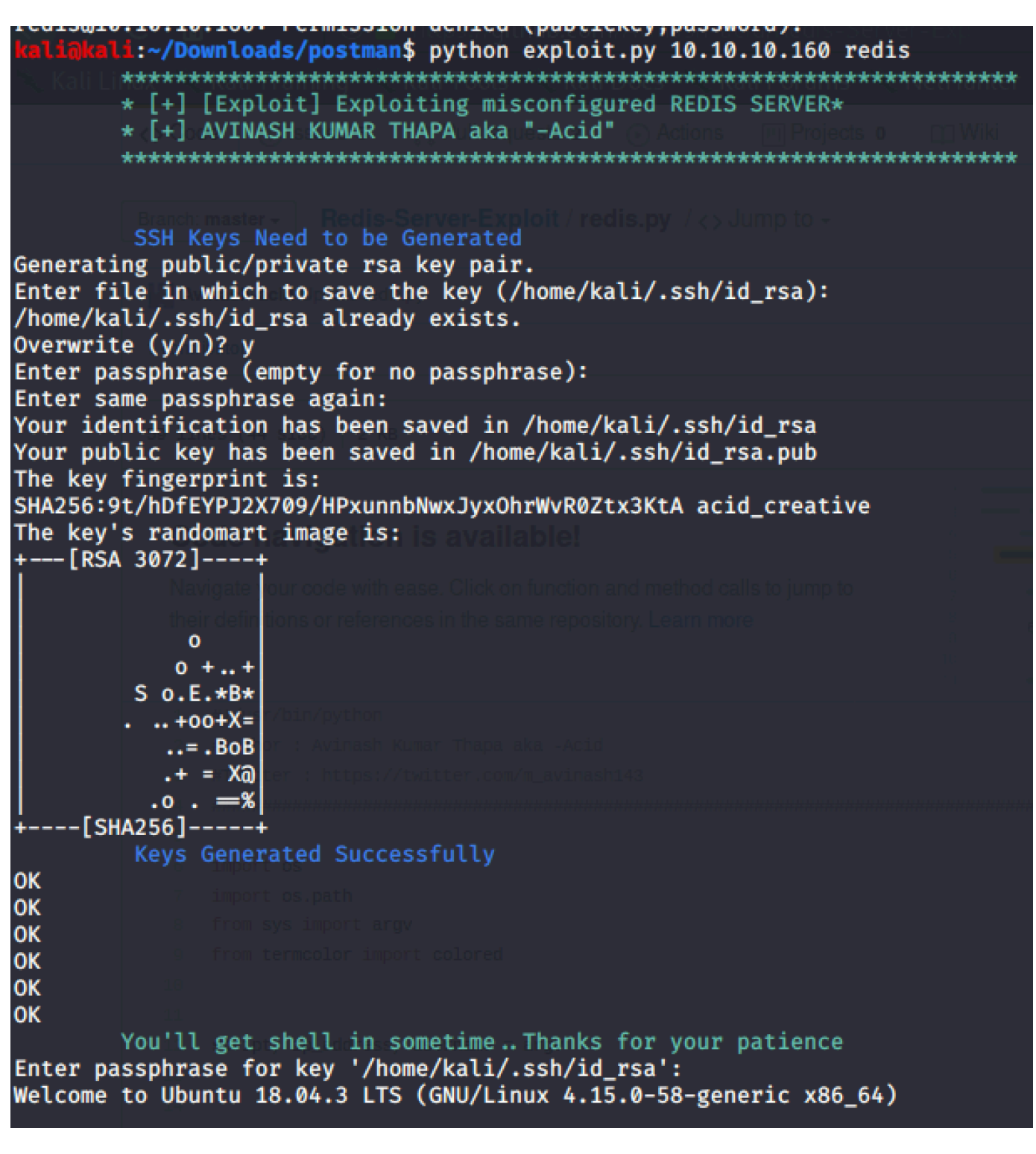


Using this guide, we can enumerate the Redis service further: https://averagesecurityguy.github.io/code/pentest/2015/09/17/pentesting-redis-servers/

Redis Exploit

In essenece, this exploit required you to workout that seeing *'authorized-keys'* in the lists that the `redis-cli` offered meant that it would be possible to locate `/var/lib/redis/.ssh/` to exploit. https://medium.com/@Victor.Z.Zhu/redis-unauthorized-access-vulnerability-simulation-victor-zhu-ac7a71b2e419 .

I had a lot of trouble getting these exploits to work, but then I found this exploit on github: https://github.com/fagray/redis-exploit . Ensure you change the path of the exploit to go to `"/var/lib/` ", and give the exploit a password of at least 8 chars, or it wont work.



Redis Shell

We can't get our user flag just yet. That belongs to someone called **Matt**. We need to enumerate around the box.

In the `/opt/` folder, there's a private key. We need to give that to `ssh2john` back in kali, and then crack it through `John` .

Copy and save this into a text file on your kali. I called mine `private` .

Then use `/usr/share/john/ssh2john.py private > privatejohn` , which outputs a hash that John can crack.

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

Go back to the victim shell, and `su Matt` , and give the password `computer2008` . You will then have a **Matt Shell**

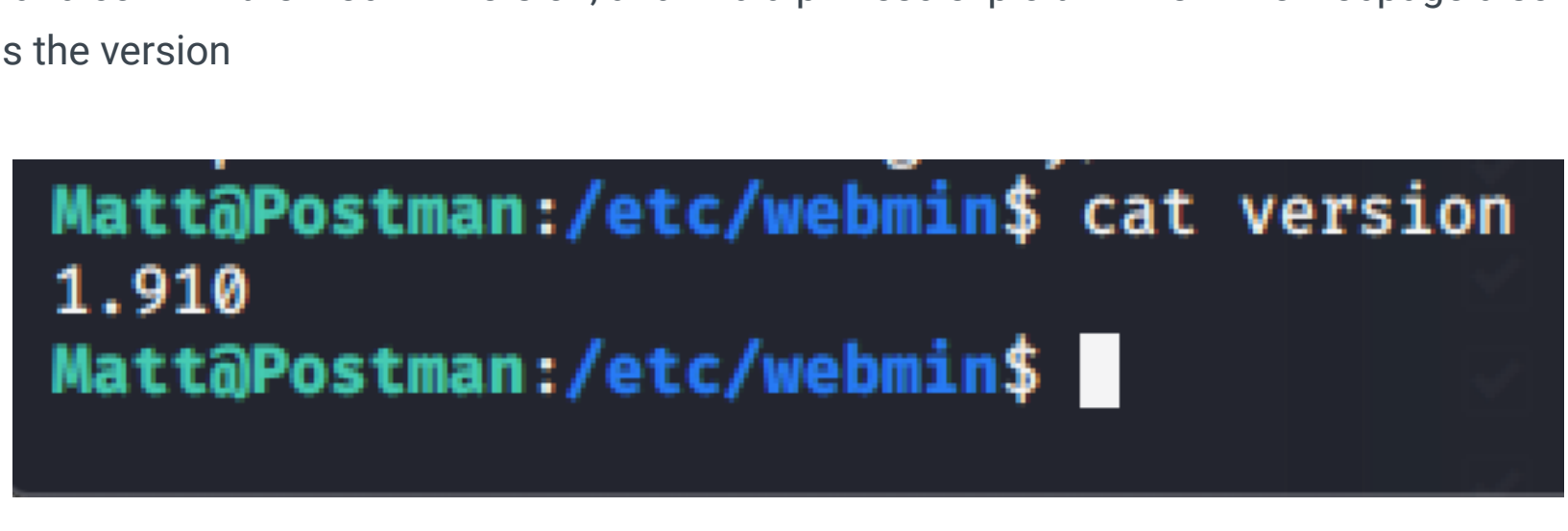
I also wondered if these creds get us into the `webmin` login screen back on port `10000` , and it does!

Matt Shell

Transfer over your favourite enumeration scripts, and let's get to work.

Straight away, something to do with root and webmin comes up. This is likely the exploit route.

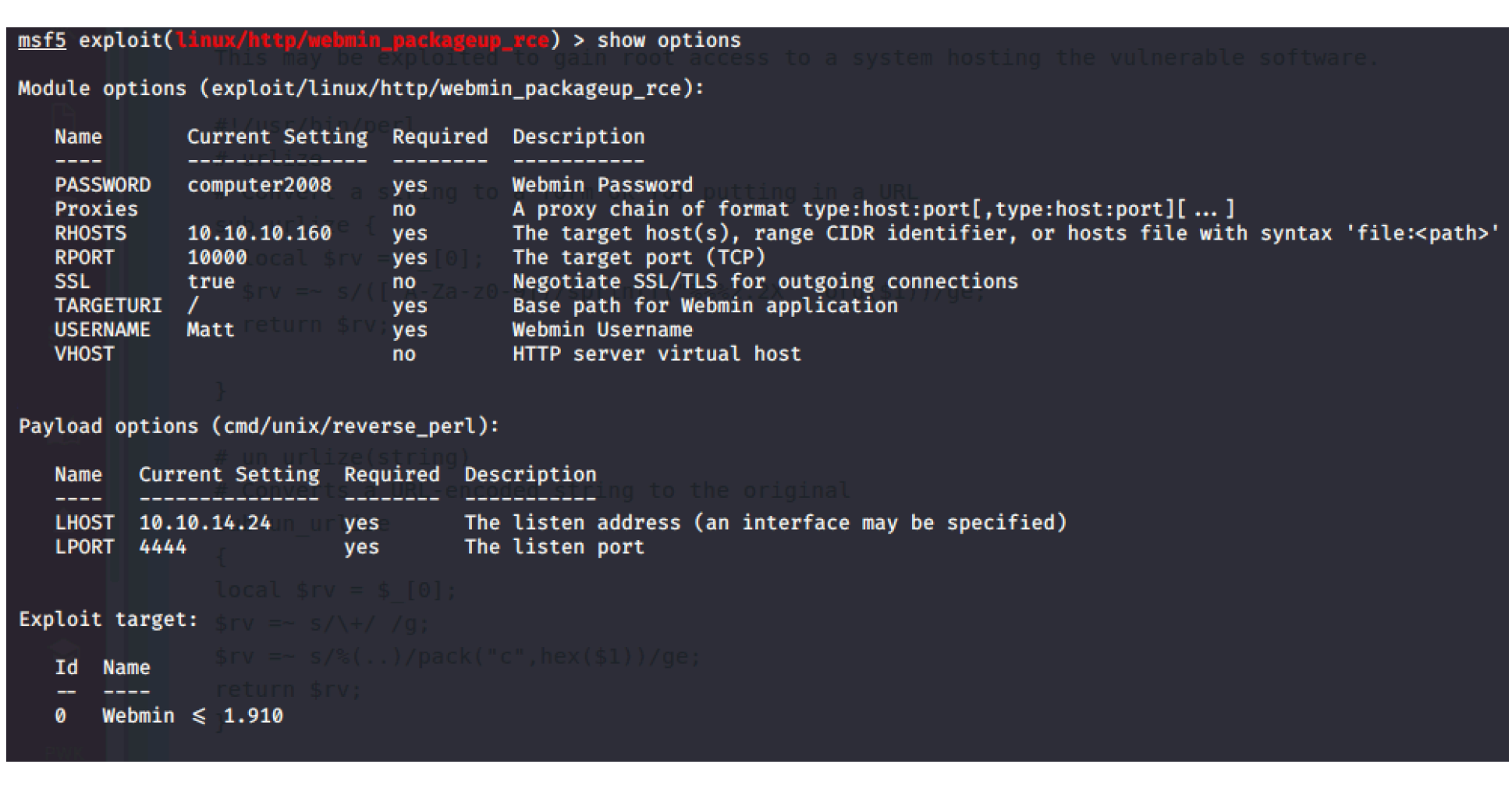
Let's go and confirm the webmin version, and find a priv esc exploit: **1.910 .** The webpage also confirms the version



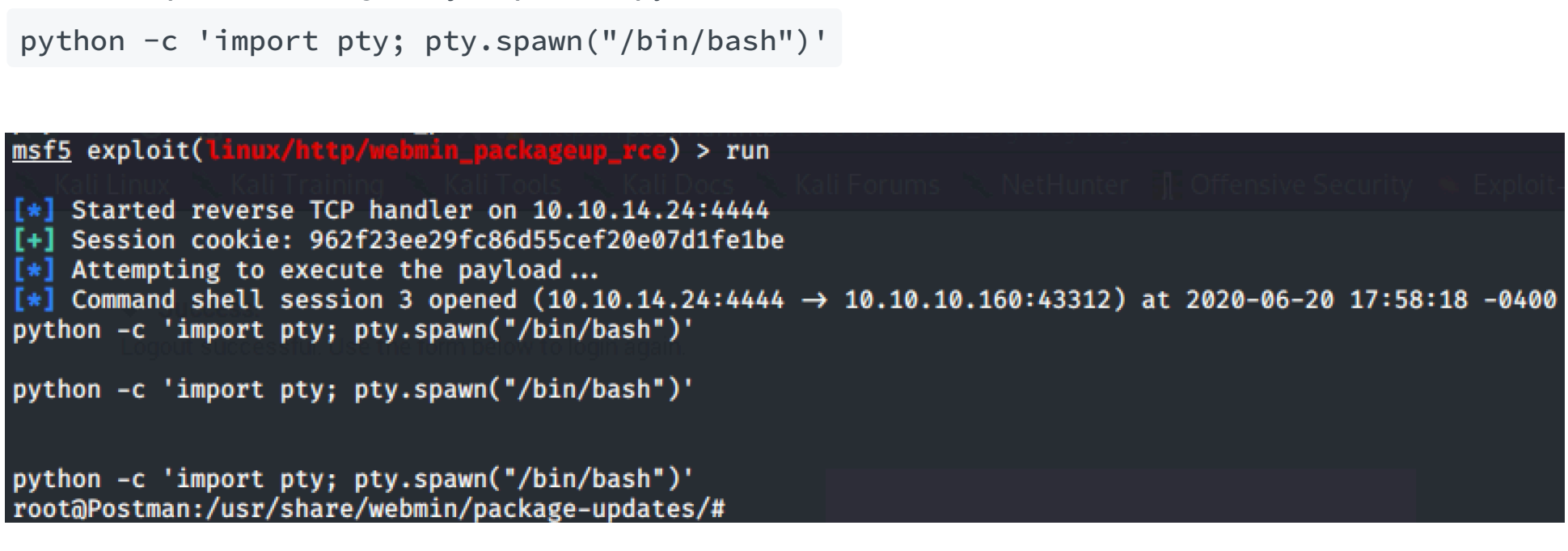
Webmin - Metasploit

If we search for *webmin 1.910 priv esc*, we'll find this link https://www.exploit-db.com/exploits/46984 that details a metasploit exploit.

Load up `msfconsole` , and then `use linux/http/webmin_packageup_rce` . Input the necessary options



The metasploit will hang, but just put in a python interactive shell to retrieve it:
`python -c 'import pty; pty.spawn("/bin/bash")'`



You're root! Go and do rooty things!