Linux & Python Project No.1
SDA Academy

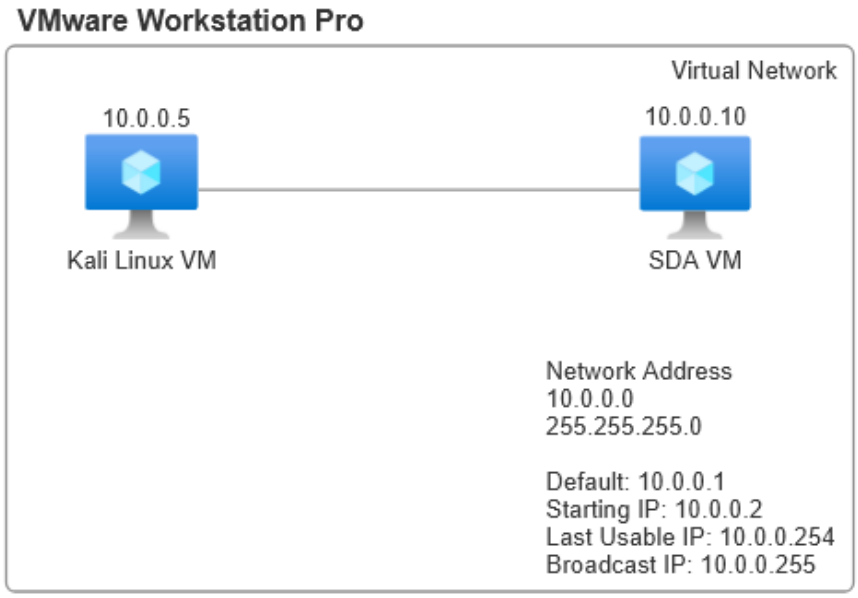
Report for the SDA.vm and the automation process of enumeration and password cracking with Python.

Group members:
Geri
Dean
Marvin

Important!

Due to the virtual machine being incompatible with the VMware and having problems with the vm's network interface not getting the IP address from the virtual private network it only worked on Bridge Interface on Virtual Box where as Kali was on NAT-ed virtual interface on VMware, anyways this way it worked and we're going to proceed this way.

Topology of the Lab



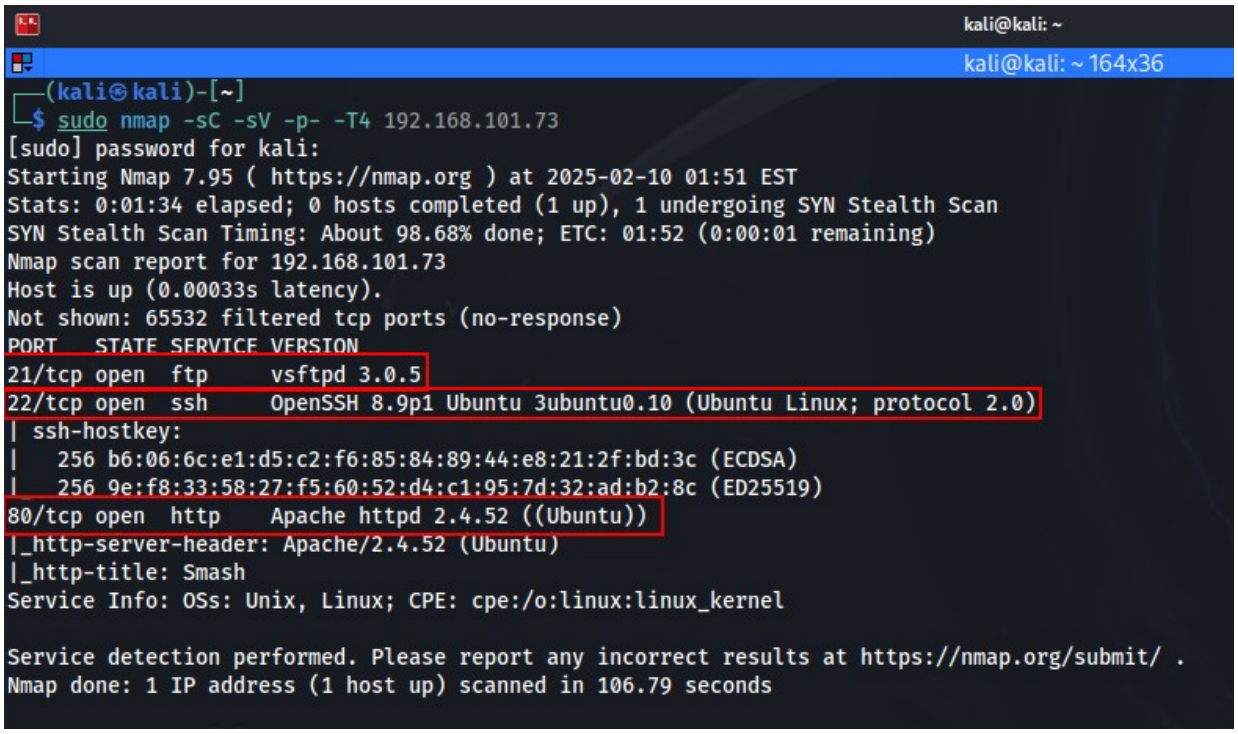
As explained above we're going to work with different hypervisor platforms to "hack" the SDA.vm manually first and then automate it, on automation process we'll use this topology to not confuse the script and find the exact IP of the machine.

Recon & Enumeration Phase

-Nmap Scan (Checking for open ports and service versions)

Command
sudo nmap -sS -sV -p- -T4 192.168.0.2

Output

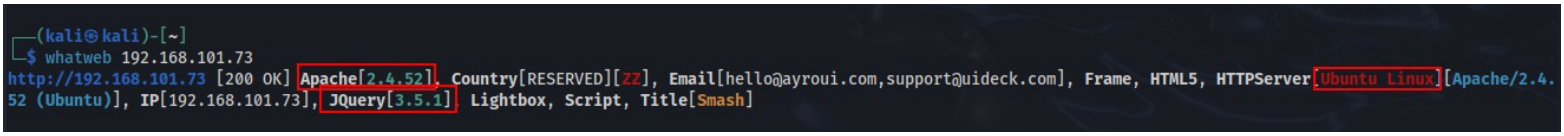


A short explanation of the output, so we have 3 open ports, If we would go through 21 (FTP) it would ask for a password, usually FTP has Anonymous login allowed on vulnerable VM to make it interesting this doesn't same thing goes for SSH and rarely are vulnerabilities on this service brutefirce is the only way to get over it but it's a waste of time when you don't have more info on users. The last one port 80 (Apache) or Web application may have misconfigurations or some other kind of webapp vulnerabilities.

I’m going to give it a whatweb to see if there are any frameworks.

Command
whatweb 192.168.101.73

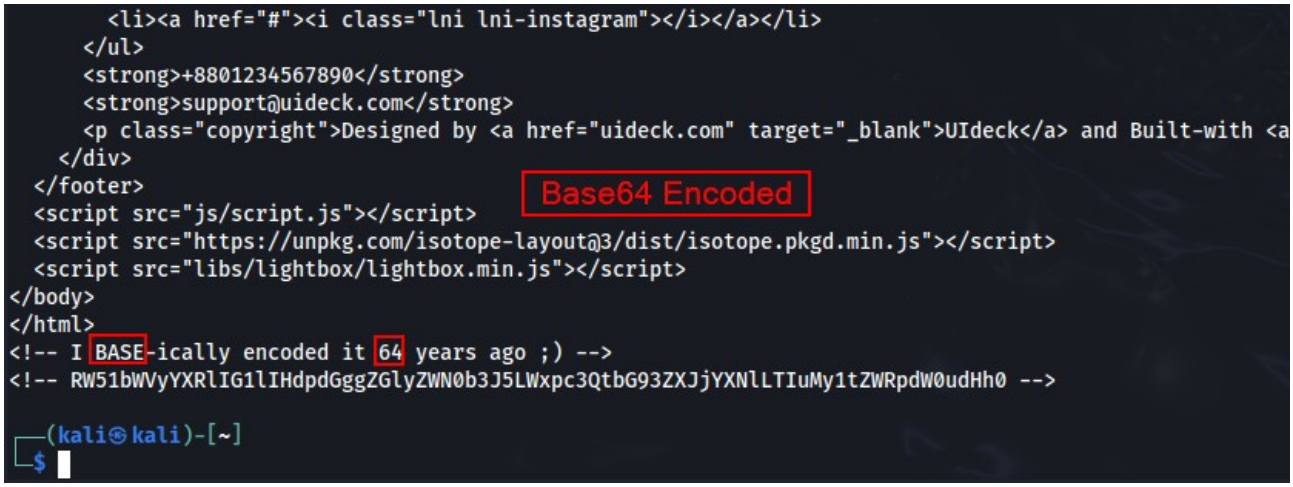
Output



After that I gave the curl command to view the page source code for comments or functions.

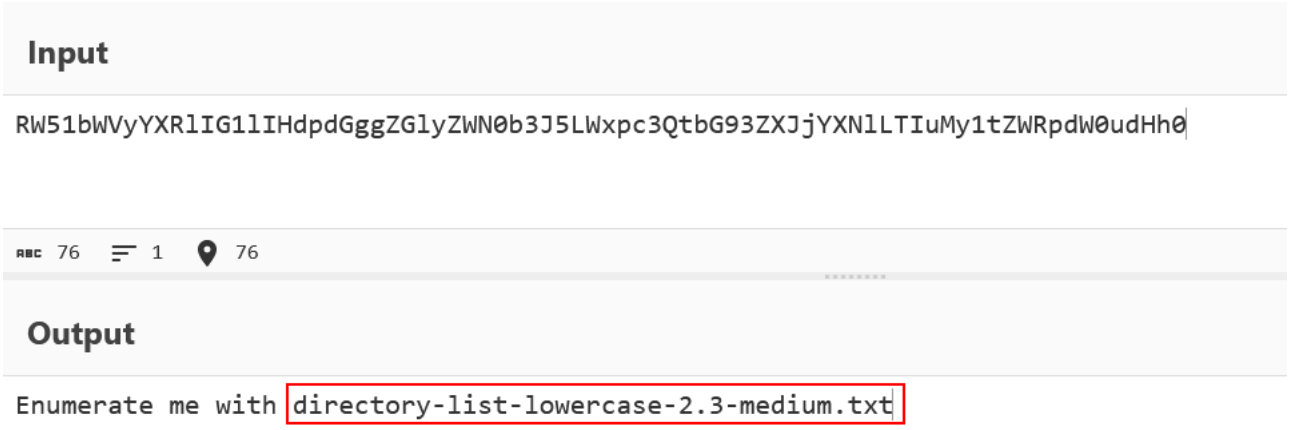
Command
curl 192.168.101.73

Output

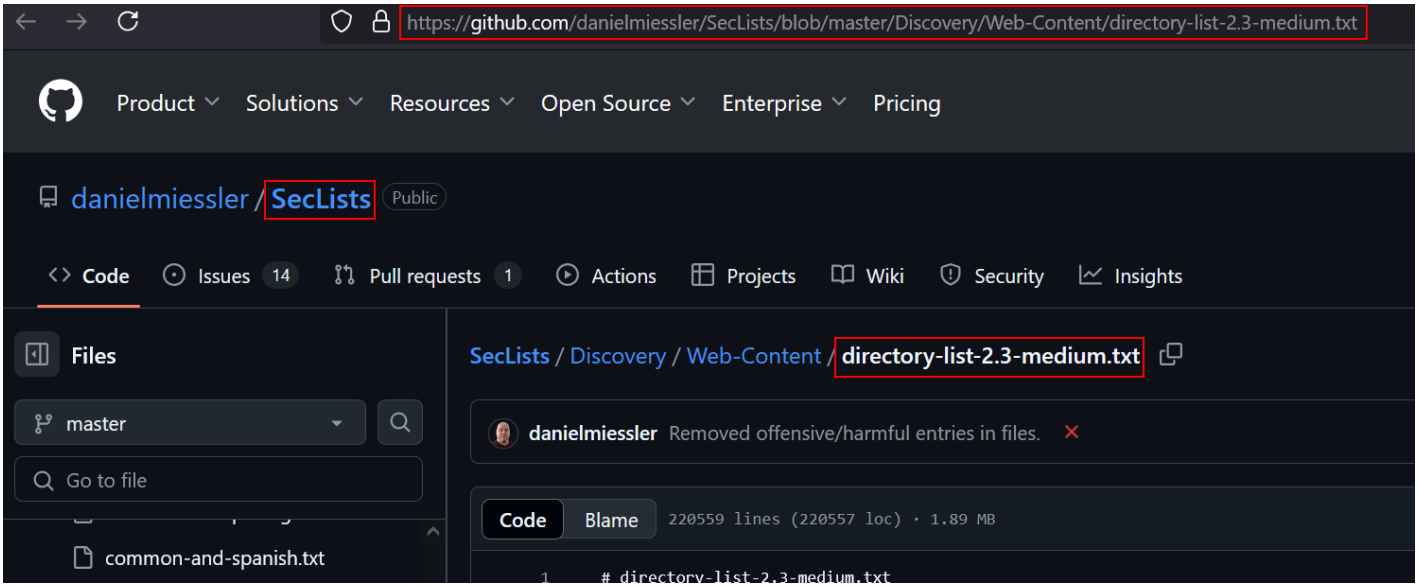


As soon as I gave the curl command at the end of the code I see the message, directly noticed the BASE and number 64 immediately went to decode it.

Decoding the Base64 code that was found in web's source code



Message gives us details on how to enumerate the webapp.

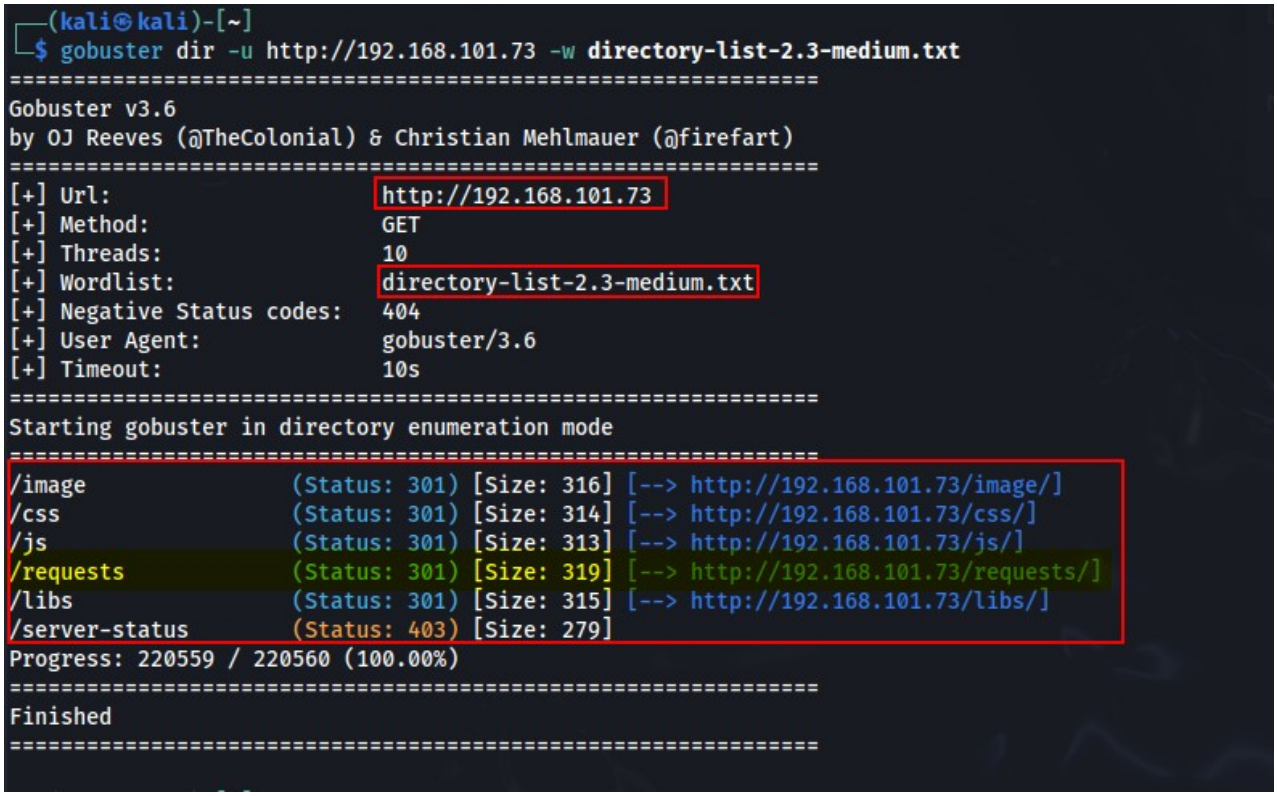


I proceeded to download the list and used gobuster to look for directories.

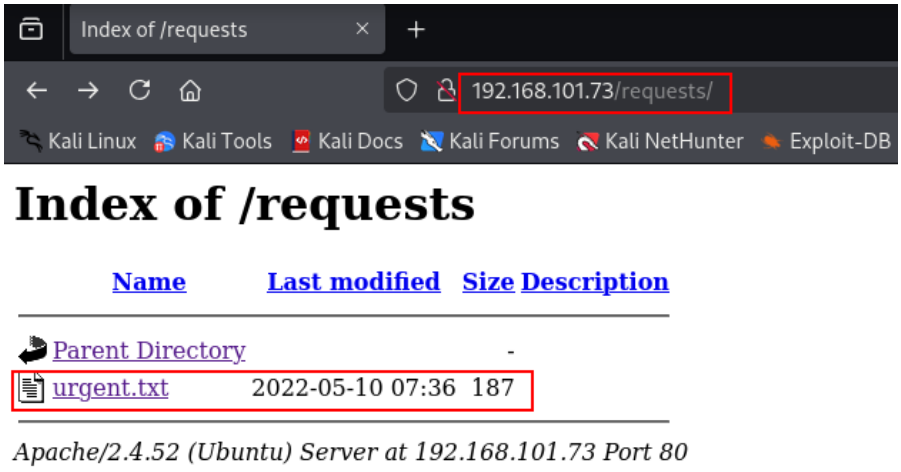
GoBuster brute-forcing Directories

Command
gobuster dir -u http://192.168.101.73 -w directory-list-2.3-medium.txt

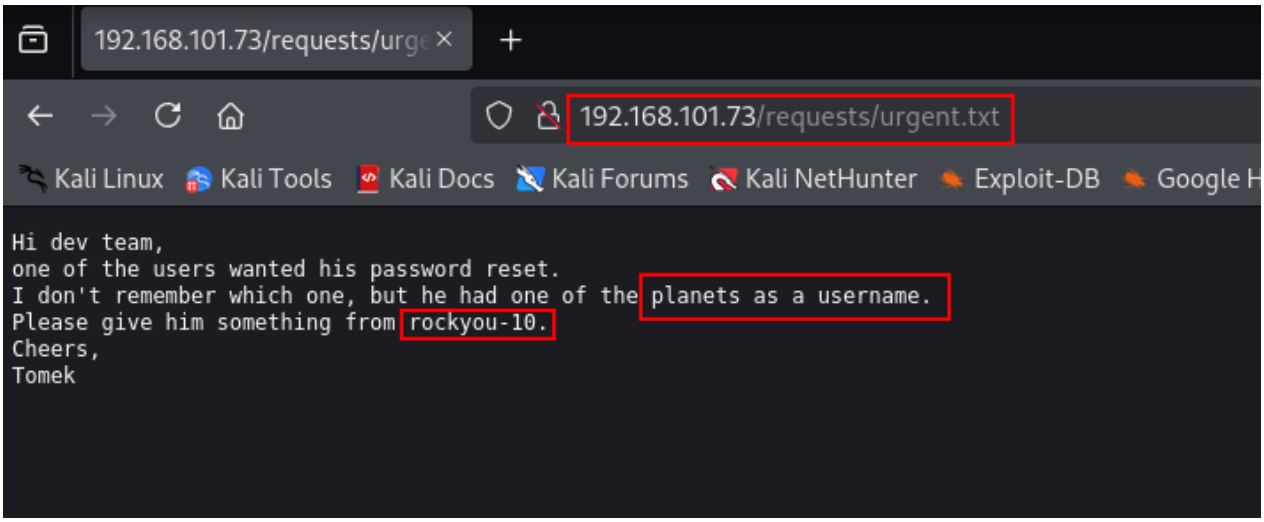
Output



I checked all directories that were found with GoBuster and the most interesting one was **/requests** directory.

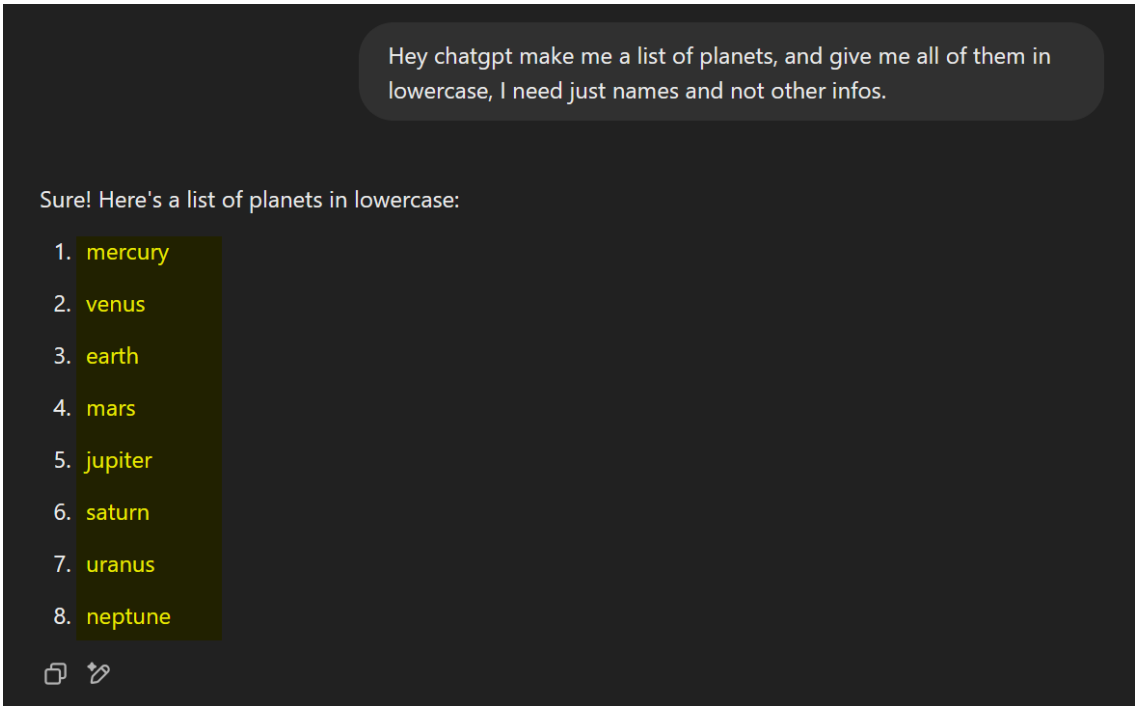


urgent.txt



Here we see that the user has as username a planet name, we're going to make a list of planets using ChatGPT then put them on a list and use hydra to Brute-Force on the SSH service.

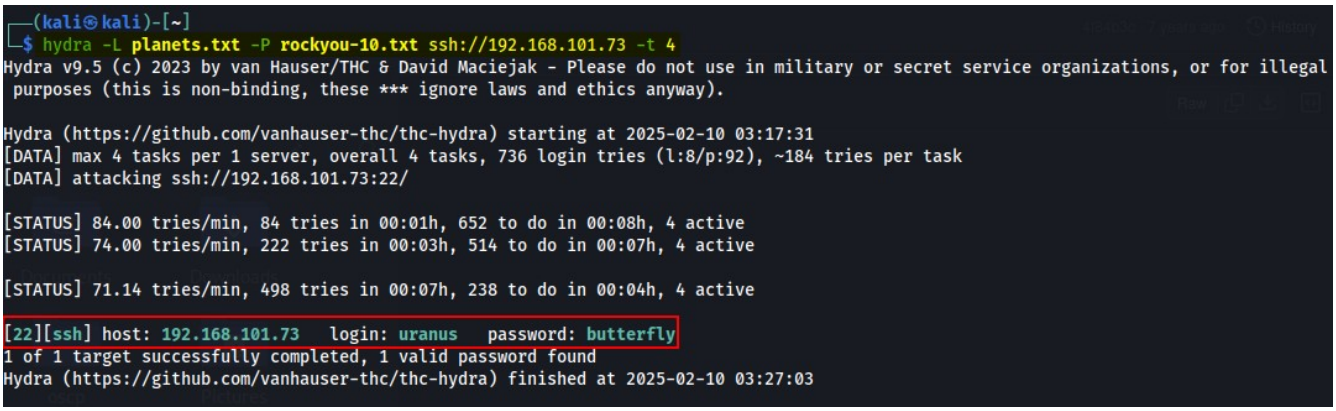
Also yes we're going to use the rockyou-10 as passwords list, it's short and it's also shown on the urgent.txt message.



Brute-Forcing SSH Logins using Hydra

Command
hydra -L planets.txt -P rockyou-10.txt ssh://192.168.101.73 -t 4

Output

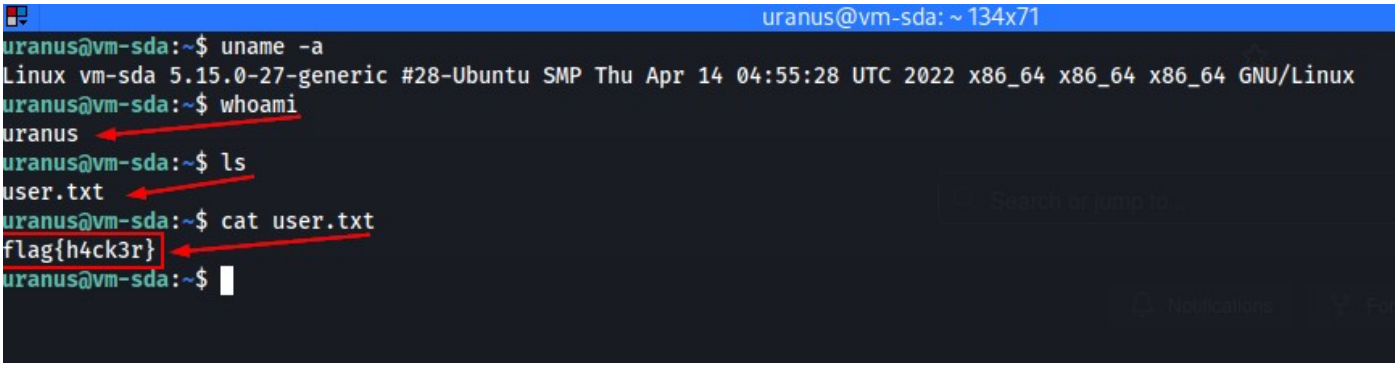


Credentials

uranus:butterfly

Since we found the credentials of the user, we're going to login with SSH and see for possible privilege escalation ways to get into root.

We logged in on uranus user and found the first flag: **user.txt**.



Flag: **flag{h4ck3r}**

Privilege Escalation to root user.

Now that we are logged in we're going to see what permissions as uranus user we have using sudo -l command.

Command
sudo -l

Output

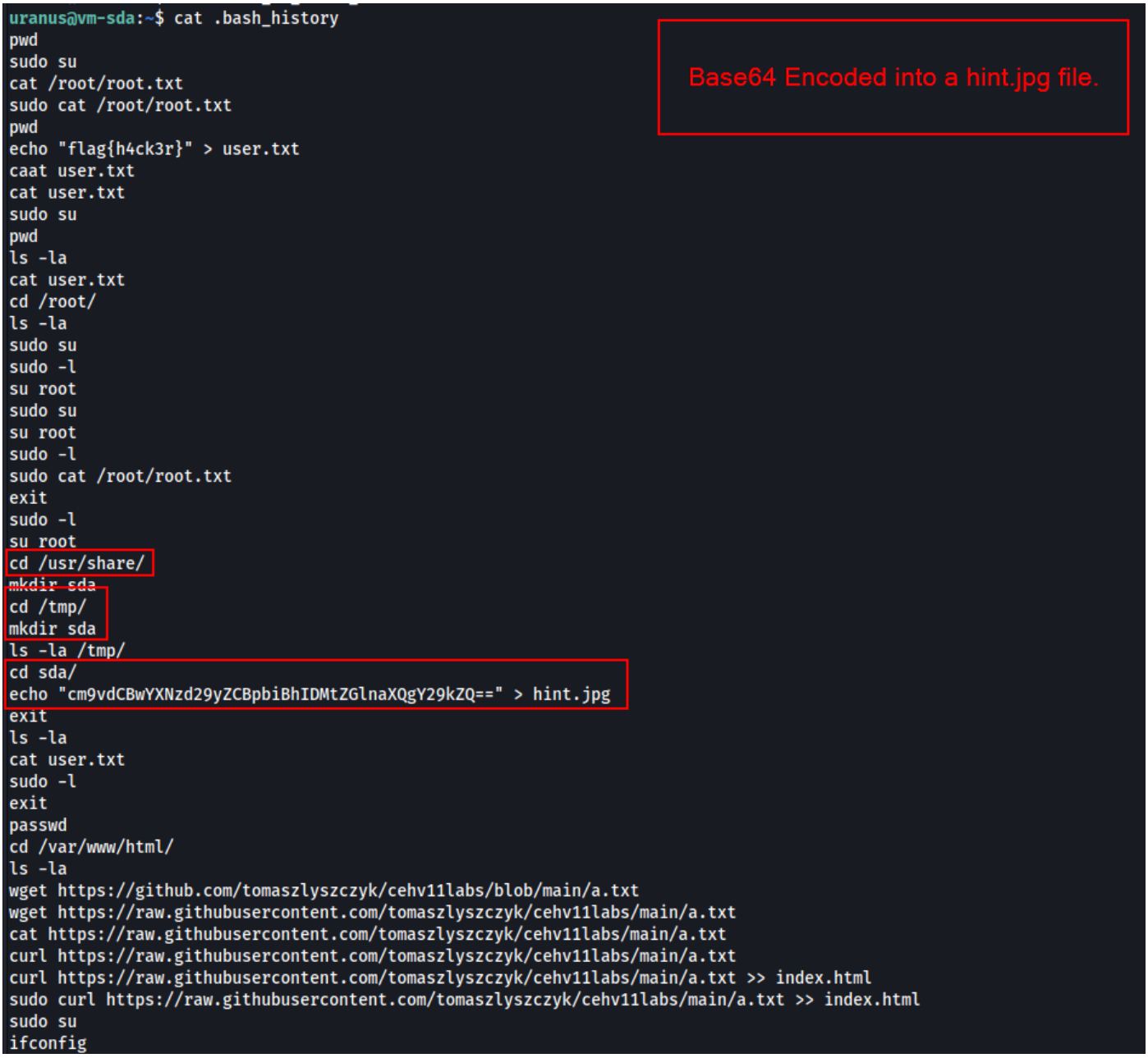
```
uranus@vm-sda:~$ sudo -l
[sudo] password for uranus:
Sorry, user uranus may not run sudo on vm-sda.
uranus@vm-sda:~$
```

Looks like we don't have permissions to run sudo command with uranus, usually we I get into vulnerable machines or have access to a user with low privileges I try to see their last commands they've put on the machine.

I managed to find some good findings from the bash history.

```
uranus@vm-sda:~$ ls -lash
total 40K
4.0K drwxr-x--- 4 uranus uranus 4.0K May 10  2022 .
4.0K drwxr-xr-x 3 root   root   4.0K May 10  2022 ..
4.0K -rw------- 1 uranus uranus 1021 Feb  6 11:41 .bash_history
4.0K -rw-r--r-- 1 uranus uranus  220 Jan  6  2022 .bash_logout
4.0K -rw-r--r-- 1 uranus uranus 3.7K Jan  6  2022 .bashrc
4.0K drwx------ 2 uranus uranus 4.0K May 10  2022 .cache
4.0K -rw-r--r-- 1 uranus uranus  807 Jan  6  2022 .profile
4.0K drwx------ 2 uranus uranus 4.0K May 10  2022 .ssh
   0 -rw-r--r-- 1 uranus uranus    0 May 10  2022 .sudo_as_admin_successful
4.0K -rw-rw-r-- 1 uranus uranus   13 May 10  2022 user.txt
4.0K -rw-rw-r-- 1 uranus uranus  215 May 10  2022 .wget-hsts
```

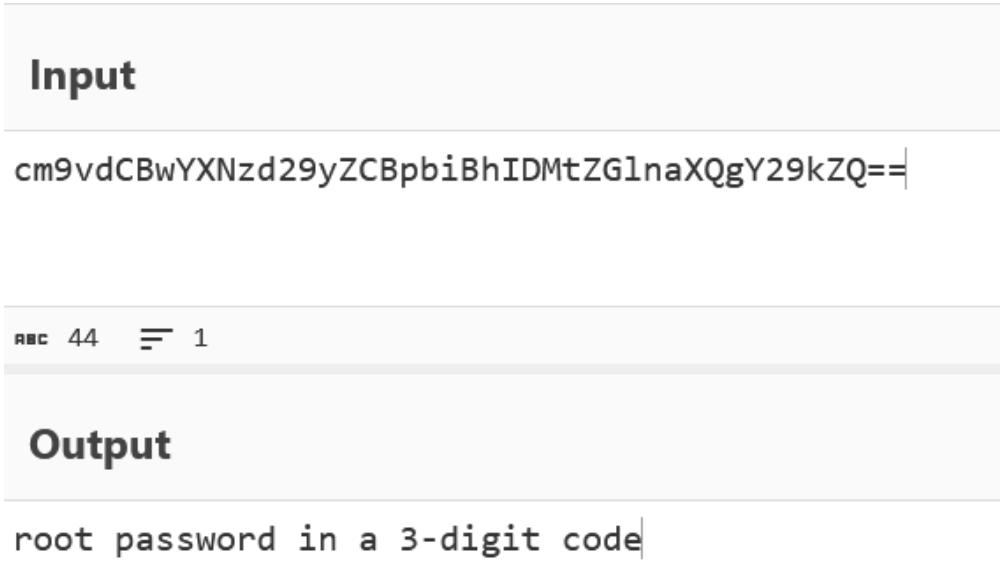
Accessing the .bash_history file.




```
uranus@vm-sda:~$ cat .bash_history
pwd
sudo su
cat /root/root.txt
sudo cat /root/root.txt
pwd
echo "flag{h4ck3r}" > user.txt
caat user.txt
cat user.txt
sudo su
pwd
ls -la
cat user.txt
cd /root/
ls -la
sudo su
sudo -l
su root
sudo su
su root
sudo -l
sudo cat /root/root.txt
exit
sudo -l
su root
cd /usr/share/
mkdir sda
cd /tmp/
mkdir sda
ls -la /tmp/
cd sda/
echo "cm9vdCBwYXNzd29yZCBpbiBhIDMtZGlnaXQgY29kZQ==" > hint.jpg
exit
ls -la
cat user.txt
sudo -l
exit
passwd
cd /var/www/html/
ls -la
wget https://github.com/tomaszlyszczyk/cehv11labs/blob/main/a.txt
wget https://raw.githubusercontent.com/tomaszlyszczyk/cehv11labs/main/a.txt
cat https://raw.githubusercontent.com/tomaszlyszczyk/cehv11labs/main/a.txt
curl https://raw.githubusercontent.com/tomaszlyszczyk/cehv11labs/main/a.txt
curl https://raw.githubusercontent.com/tomaszlyszczyk/cehv11labs/main/a.txt >> index.html
sudo curl https://raw.githubusercontent.com/tomaszlyszczyk/cehv11labs/main/a.txt >> index.html
sudo su
ifconfig
```

I got another Base64 message, I'm going to decode it and see what it says.

Decoding Base64 found from .bash_history.

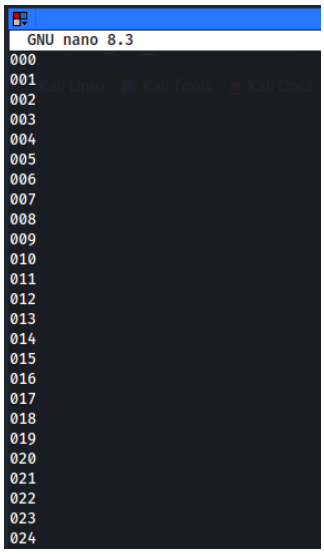


Okay so we see that the root user has a 3-digit code now I'm going to use a tool called crunch, this tool will generate a list of 3-digit codes from 000 to 999 then again use hydra tool to brute-force root user.

Using crunch to generate 3-digit codes.

Command
crunch 3 3 -t %%% -o codes.txt

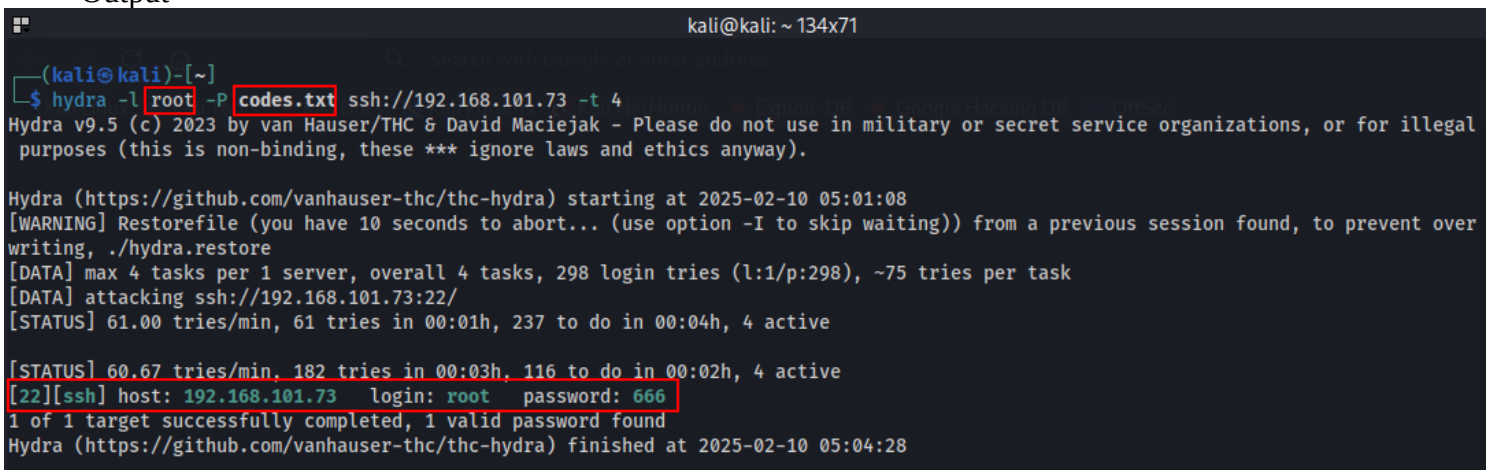
Output



Now we're going to use hydra again to brute-force root user.

Command
hydra -l root -P codes.txt ssh://192.168.101.73 -t 4

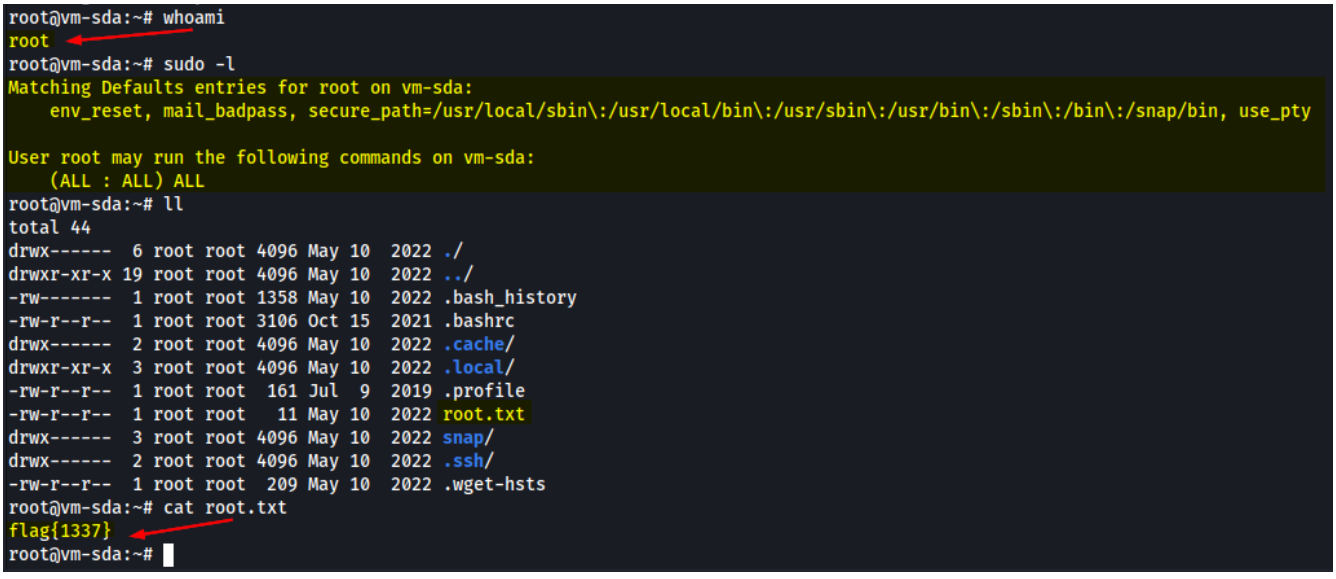
Output



Password

root:666

Accessing root user with found credentials



Flag **flag{1337}**

And with the last flag we pwned the SDA.vm machine, now to the next part where we automate this process using python3 and scapy and the script will be uploaded on a GitHub repo.