Final Engagement

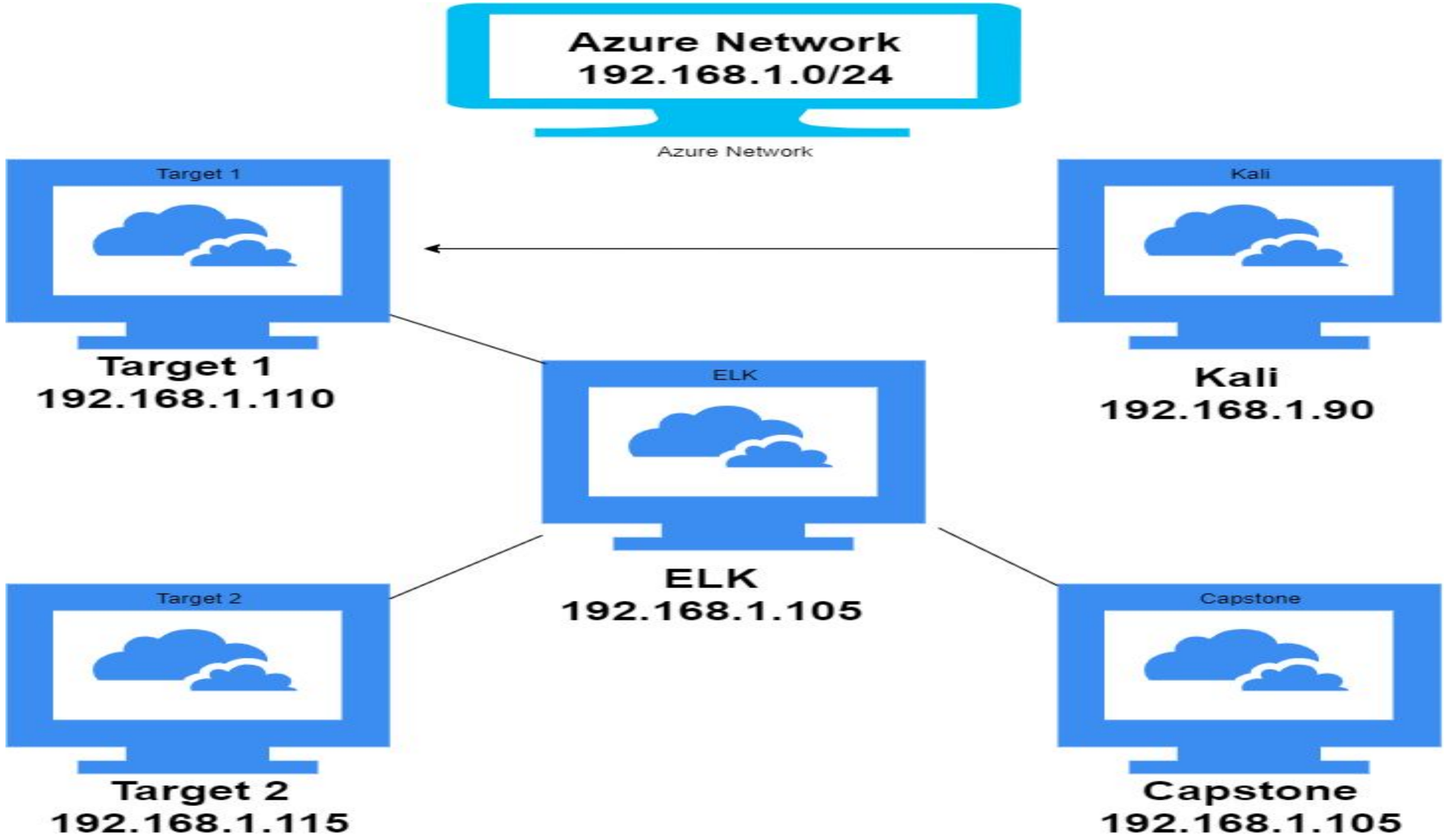
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:
192.168.1.0/24
Netmask:255.255.255.0
Gateway:192.168.1.1

Machines
IPv4:192.168.1.90
OS:Linux
Hostname:Kali

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

IPv4: 192.168.1.110
OS: Linux
Hostname: Target 1

IPv4: 192.168.1.115
OS: Linux
Hostname: Target 2

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
Weak Password	michael: michael steven:pink84	• One was simply to guess • One was easy to crack the hash
Wordpress	Passwords for SQL in plain text WPScan Provided Recon Information	• Just needed access to machine to gain additional access • Easily Able to obtain Usernames
Sudo python	Had sudo privileges for python	• Able to use python script to elevate to root access

Critical Vulnerabilities: Target 2

Our assessment uncovered the following critical vulnerabilities in **Target 2**.

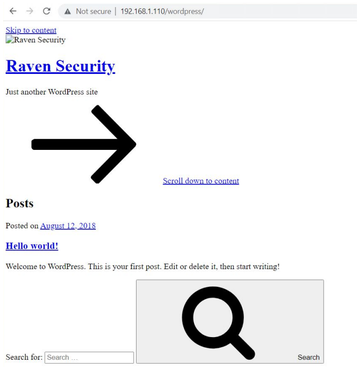
Vulnerability	Description	Impact
Hidden Files	Normally only seen by those with credentials, these directories and files weren’t locked from public viewing	Potential confidential information located here
	Takes forever	dirb http://192.168.1.115/
	Quick but had to install first	gobuster -w /usr/share/wordlist dir -u http://192.168.1.115/

Exploits Used

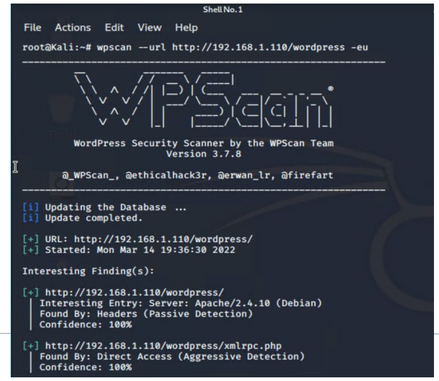
Exploitation: WordPress

Summarize the following:

- Typed “192.168.1.110/wordpress/” into browser, to confirm the server WordPress site was up and running.”



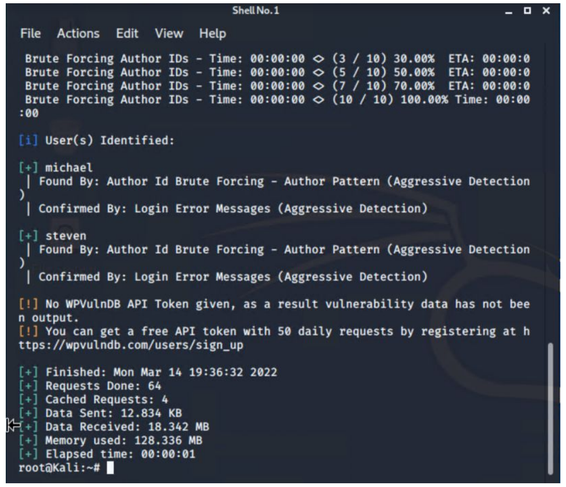
- Used the command wpscan –url http://192.168.1.110/wordpress -eu to author ID brute force potential usernames.



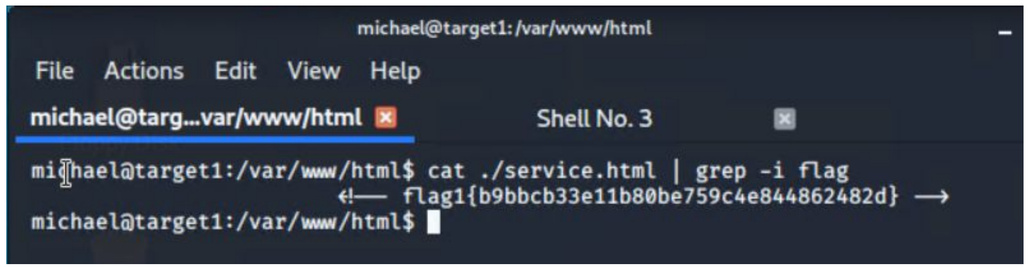
Exploitation: WordPress

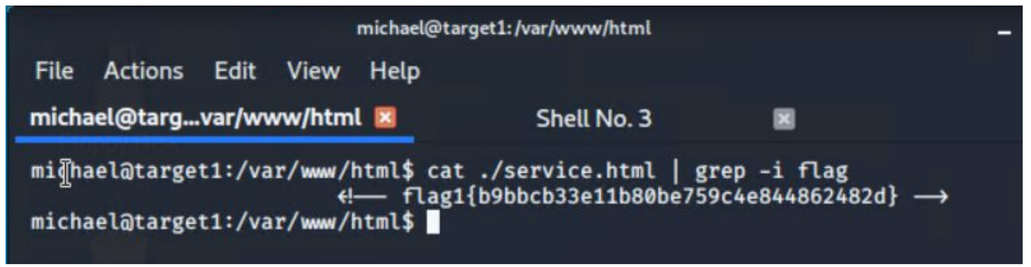
Achievements

- Was able to determine that michael and steven are users



- Was able to locate flags 1 and 2 with Michael's credentials

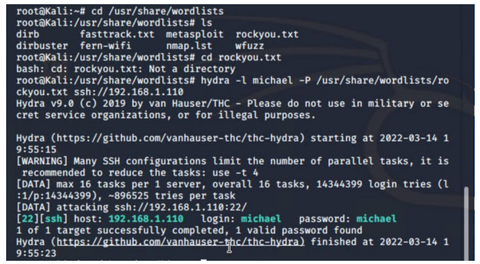




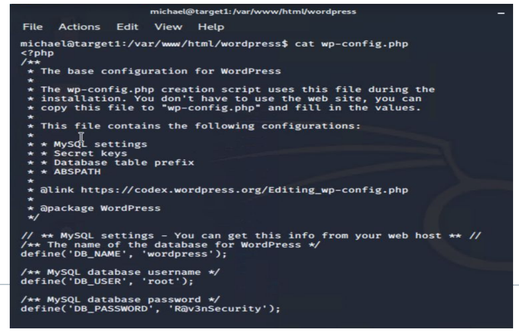
Exploitation: Weak Password

Summarize the following:

- michael:michael, password same as username, easy to guess
- Another option was to use the command hydra -l michael -P /usr/share/wordlists/rockyou.txt ssh://192.168.1.110



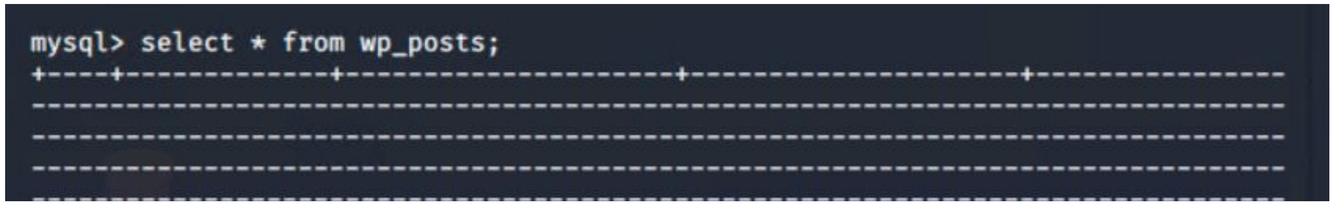
- steven: pink84 Weak password. Hash was quick to decipher using John the Ripper
- MySQL passwords were in plain text in the wp-config file (root:R@v3nSecurity)

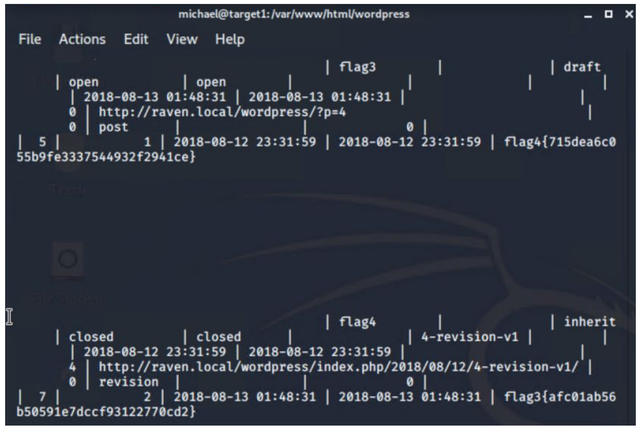


Weak Passwords

Achievements

- Was able to locate flags 3 and 4 with the saved plain text MySQL credentials

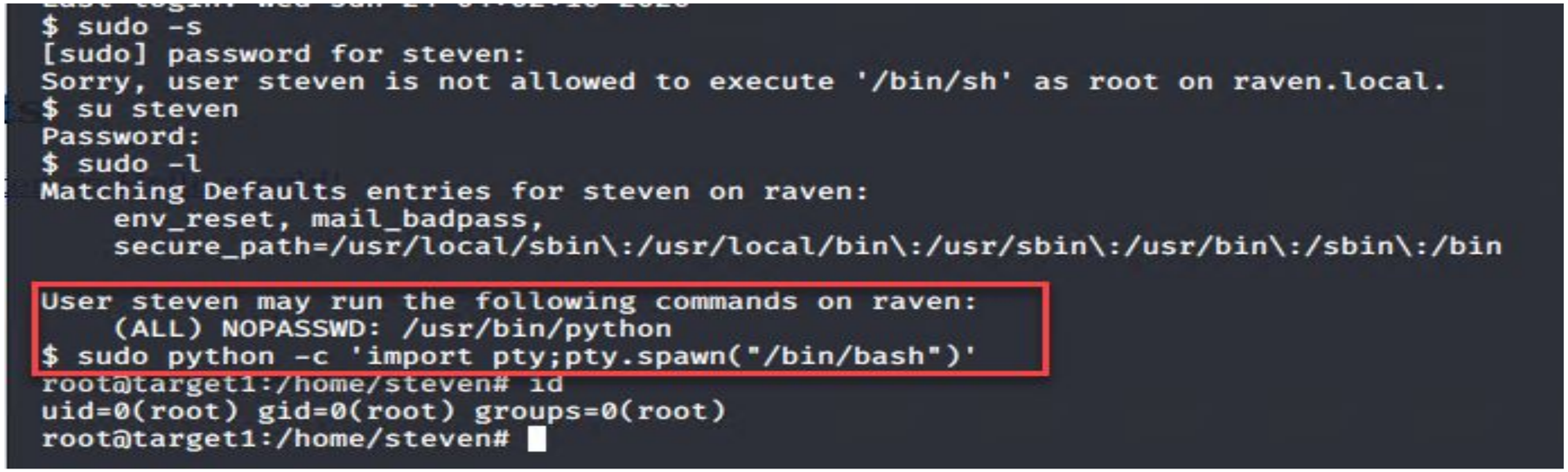




- Flags contained password hashes that were able to be cracked using John the Ripper.

Exploitation: Sudo Python

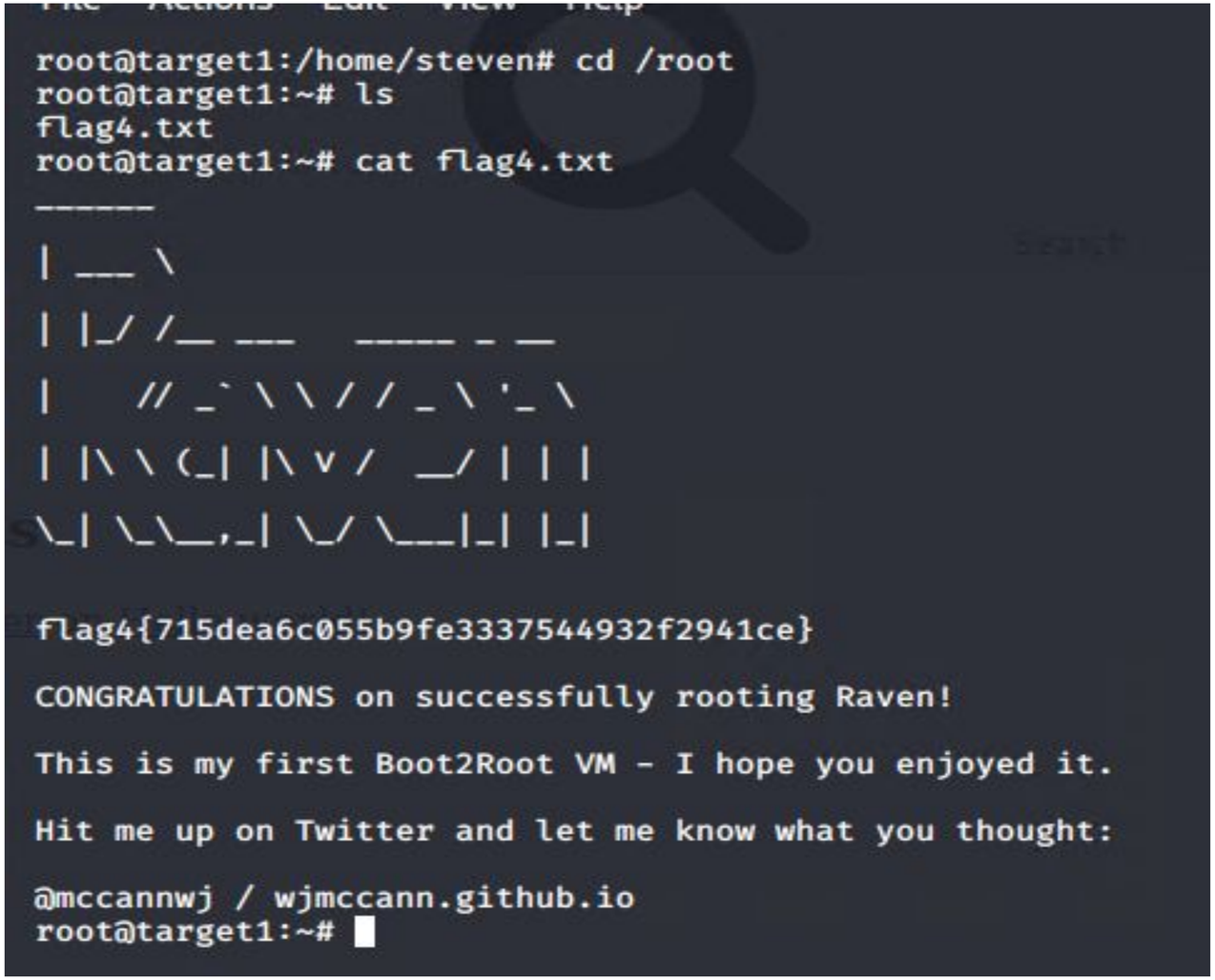
- Exploited the password hash of user '**steven**' using John the Ripper and accessed the account
  - Username: steven
  - Password: pink84
- User had python root privileges which were exploited through spawn shell.
- Command used: ***sudo python -c 'import pty;pty.spawn("/bin/bash")'***



Sudo Python

Achievements

- Was able to find flag 4 after gaining root access to steven's account
- Commands used:
  - ***cd /root***
  - ***ls***
  - ***cat flag4.txt***



Avoiding Detection

Stealth Exploitation of Wordpress (Enumeration)

Monitoring Overview

- **Which alerts detect this exploit?**

  Kibana Alert: HTTP Request Size Monitor (WHEN sum() of http.request.bytes OVER all documents IS ABOVE 3500 FOR THE LAST 1 minute)

- **Which metrics do they measure?**

  HTTP Request Size Monitor: Bytes of HTTP Requests

- **Which thresholds do they fire at?**

  HTTP Request Size Monitor: Above 3500 Bytes in 1 minute

Stealth Exploitation of Wordpress (Enumeration) Part 2

Mitigating Detection

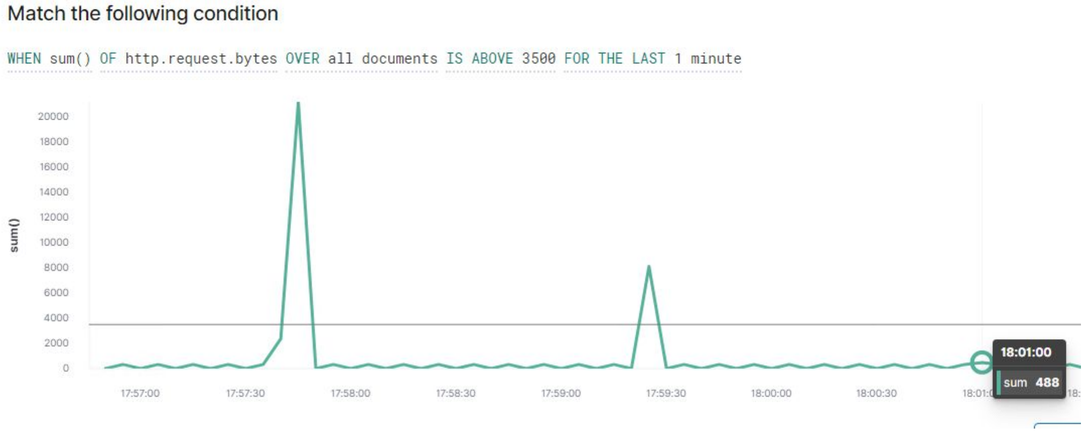
- **How can you execute the same exploit without triggering the alert?**

  Passive and mixed options are available on the WPScan, which have smaller request sizes and can potentially avoid triggering the Alerts, but they also will produce less information

- **Are there alternative exploits that may perform better?**

  Using the passive or mixed options for a scan will lower the HTTP request Size, however other methods may need to be utilized because passive WPScans will produce less reconnaissance information.

  Command: wpscan --url http://192.168.1.110/wordpress -- enumerate u --detection-mode (passive, mixed, aggressive)



Stealth Exploitation of Weak Passwords

Monitoring Overview

- **Which alerts detect this exploit?**

  Kibana Alert: CPU Usage Monitor (WHEN max() OF system.process.cpu.total.pct OVER all documents IS ABOVE 0.5 FOR THE LAST 5 minutes)

- **Which metrics do they measure?**

  System Processes Using CPU resources

- **Which thresholds do they fire at?**

  Above .5 for the last 5 minutes

```
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 79 candidates buffered for the current salt, minimum 96 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:54  3/3 0g/s 45406p/s 45406c/s 45406C/s catemax..calipoy
0g 0:00:00:55  3/3 0g/s 45434p/s 45434c/s 45434C/s 159a98..153sce
pink84           (steven)
1g 0:00:01:20 DONE 3/3 (2022-03-16 15:51) 0.01245g/s 46074p/s 46074c/s 46074C/s poslus..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~#
```

Mitigating Detection

- **How can you execute the same exploit without triggering the alert?**

  John the ripper uses significant CPU resources to crack passwords which could cause the alert to trigger if it is run on the target machine, copying hashes to your personal attacking machine and running John will prevent this, as it is using your own resources

- **Are there alternative exploits that may perform better?**

  There are alternate tools that can be used to crack hashes, for example, hashcat makes use of a system's GPU instead of CPU

Options

```
hashcat (v6.2.4) starting in help mode

Usage: hashcat [options]... hash|hashfile|hccapxfile [dictionary|mask|directory]...

- [ Options ] -

 Options Short / Long           | Type | Description                                              | Example
================================+======+==========================================================+=======================
 -m, --hash-type                | Num  | Hash-type, references below (otherwise autodetect)       | -m 1000
 -a, --attack-mode              | Num  | Attack-mode, see references below                        | -a 3
 -V, --version                  |      | Print version                                            |
 -h, --help                     |      | Print help                                               |
     --quiet                    |      | Suppress output                                          |
```

Stealth Exploitation of Sudo Python

Monitoring Overview

- According to the Kibana data, the alerts set were not triggered by the sudo python command. However, using a security monitoring software such as Insider Threat Management (ITM) provides alerts that are triggered by a user using a sudo command to open a root shell without a required password. If we were to have had an ITP alert to monitor sudo commands in a shell, it would have been triggered.
- Kibana monitors the ***system.auth.sudo.command*** which logs the users and the sudo commands that the user has executed where we see user ***steven*** has executed ***/usr/bin/python -c import pty;pty.spawn("/bin/bash")*** to gain root access

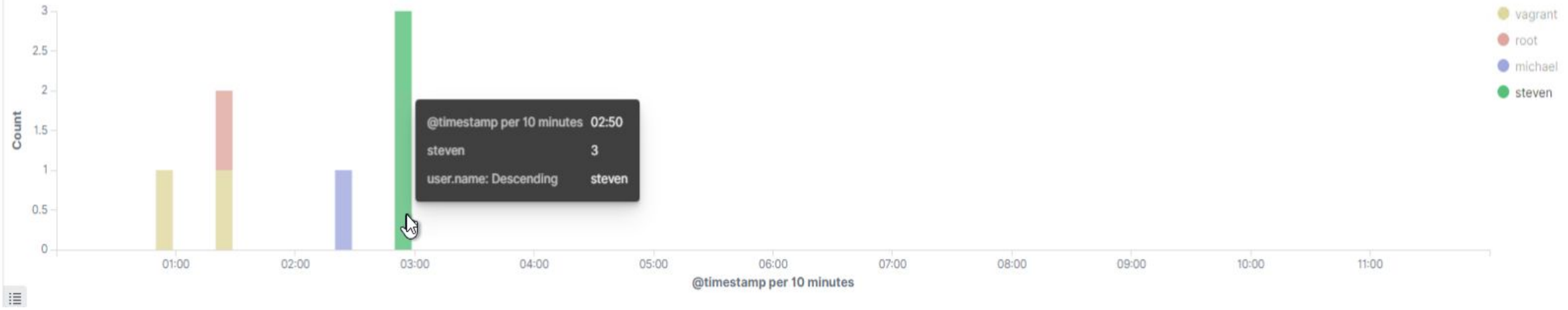
Stealth Exploitation of Sudo Python

Top sudo commands [Filebeat System] ECS

system.auth.sudo.command: Descending	user.name: Descending	Count
/bin/bash	vagrant	2
/bin/sh	steven	1
/usr/bin/apt-get install nmap -y	root	1
/usr/bin/python -c import pty;pty.spawn("/bin/bash")	steven	1
apt-install john	michael	1

Export: Raw Formatted

Sudo commands by user [Filebeat System] ECS



Stealth Exploitation of Sudo Python

Mitigating Detection

- Being that the alerts we created through Kibana were NOT triggered we wouldn’t have to change the execution of our exploit.
- If we were to implement the ITM alert discussed earlier it would be difficult to go undetected. We are unsure if it would be possible to execute the same exploitation without using sudo.

Sources:

param373r. (2020, June 10). *John the ripper*. Medium. Retrieved March 19, 2022, from https://param373r.medium.com/john-the-ripper-acf598f30ffa

Hashcat advanced password recovery. hashcat [hashcat wiki]. (n.d.). Retrieved March 19, 2022, from https://hashcat.net/wiki/doku.php?id=hashcat

Brown, K. (2020, December 30). *Use WPSCAN to scan WordPress for vulnerabilities on Kali*. Linux Tutorials - Learn Linux Configuration. Retrieved March 19, 2022, from https://linuxconfig.org/use-wpscan-to-scan-wordpress-for-vulnerabilities-on-kali

25, M., & Friedlander, G. (2021, July 26). *7 ways proofpoint ITM can detect privileged escalations in unix: Proofpoint us*. Proofpoint. Retrieved March 19, 2022, from https://www.proofpoint.com/us/blog/insider-threat-management/7-ways-proofpoint-itm-can-detect-privileged-escalations-unix